



Henning Mergard, \*1995  
Jannes Menck, \*1996

Schule:  
Vincent-Lübeck-Gymnasium, Stade

Eingang der Arbeit:  
Januar 2012

Zur Veröffentlichung angenommen:  
Februar 2012

## Sprich nur ein Wort!

### Eine universelle Sprachsteuerung für beliebige mechatronische und elektrische Geräte und Hilfsmittel

Es ist ein langer Menschheitstraum, selbst mit Geräten und Maschinen sprechen zu können. Mithilfe von Sprachsteuerung ist dies möglich geworden. Verwendet wird sie trotzdem selten, da bisher immer nur für einzelne Geräte eine Sprachsteuerung entwickelt wurde, und das ist teuer. Daher haben wir eine universelle Sprachsteuerung entwickelt, an die jederzeit neue Geräte angekoppelt werden können.

#### 1 Einleitung

Sprachsteuerungen werden bisher eher selten verwendet: So gibt es eine Sprachsteuerung, die in das Betriebssystem eines Computers integriert ist, um Texte zu diktieren. Auch in Navigationsgeräten für Autos werden Sprachsteuerungen verwendet, damit der Fahrer bei der Bedienung das Auto weiterhin unter Kontrolle hat.

Sprachsteuerung bietet entscheidende Vorteile: Der Benutzer kann auch wesentlich größere Systeme verwalten, da nur ein Mikrofon benötigt wird. Die Bedienung wird erleichtert, da der Benutzer das, was er erreichen möchte, einfach aussprechen kann. Der Benutzer kann während der Bedienung andere Dinge parallel machen oder den Standort frei verändern. Zusätzlich gibt es dem System eventuell eine menschliche Note, wenn dies auf die Sprache reagiert und

sogar antwortet. Sprachsteuerungen werden generell für das zu bedienende Gerät programmiert und quasi „maßgeschneidert“ - sie sind nicht einfach so für andere Geräte anwendbar. Wenn man nun mehrere verschiedene Geräte per Sprache steuern möchte, muss eine komplett neue Sprachsteuerung geschrieben werden, bzw. für jedes Gerät eine eigene Sprachsteuerung eingesetzt werden. Wir haben deshalb nach Möglichkeiten gesucht, eine Art universelle Sprachsteuerung zu verwirklichen, mit der es möglich ist, beliebige Geräte zu bedienen. Dabei soll die Anbindung von Geräten möglichst einfach und die Steuerung an sich möglichst kostengünstig sein.

#### 2 Projektaufbau

Mit einer Software basierten speicherprogrammierbaren Steuerung (Soft-SPS) werden in der Industrie Maschi-

nen und Anlagen gesteuert. Mit dieser lassen sich aber auch praktisch alle elektrischen, mechatronischen und mechanischen Geräte steuern. Unsere Idee war es, diese Steuerung mit einer üblichen Spracherkennungssoftware zu verknüpfen, damit man so in der Lage ist, beliebige Geräte über die Soft-SPS parallel steuern zu können. Nun benötigten wir noch eine Verbindung zwischen der Spracherkennungssoftware und der Soft-SPS. Wir wollten handelsübliche Komponenten verwenden, damit unsere Sprachsteuerung so universell und so verbraucherfreundlich wie möglich wird.

Unser System ist dementsprechend in verschiedene Abschnitte einzuteilen (siehe Abb. 1): Der erste ist die Spracherkennungssoftware. Der zweite Teil ist die Java-Bridge, die Verbindung von

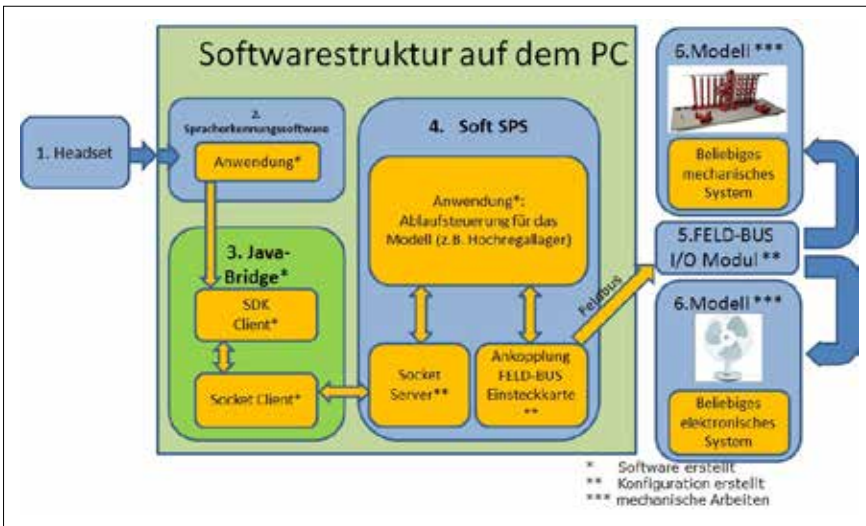


Abb. 1: Aufbau des Gesamtsystems.

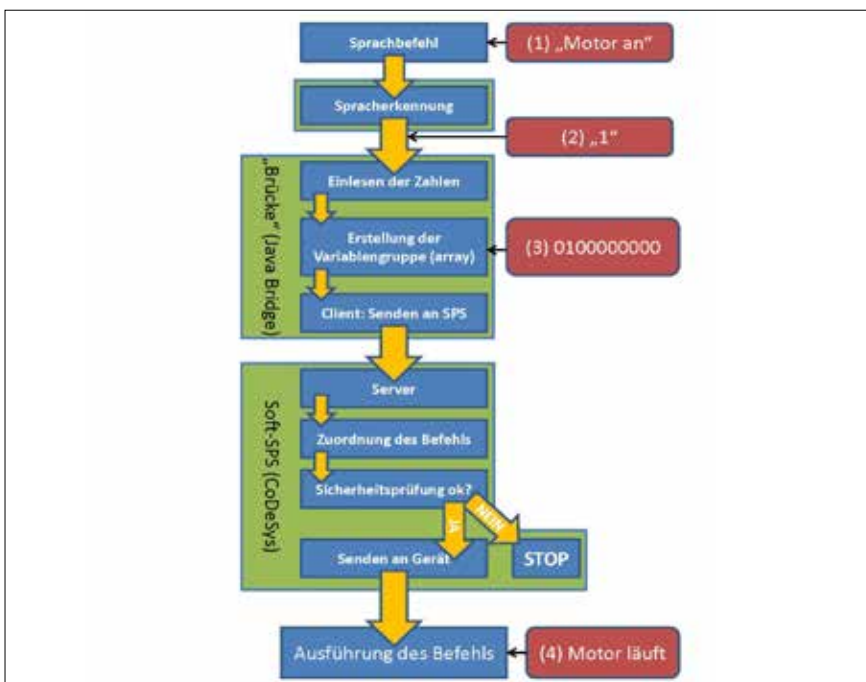


Abb. 2: Datenfluß für eine konkrete Aktion.

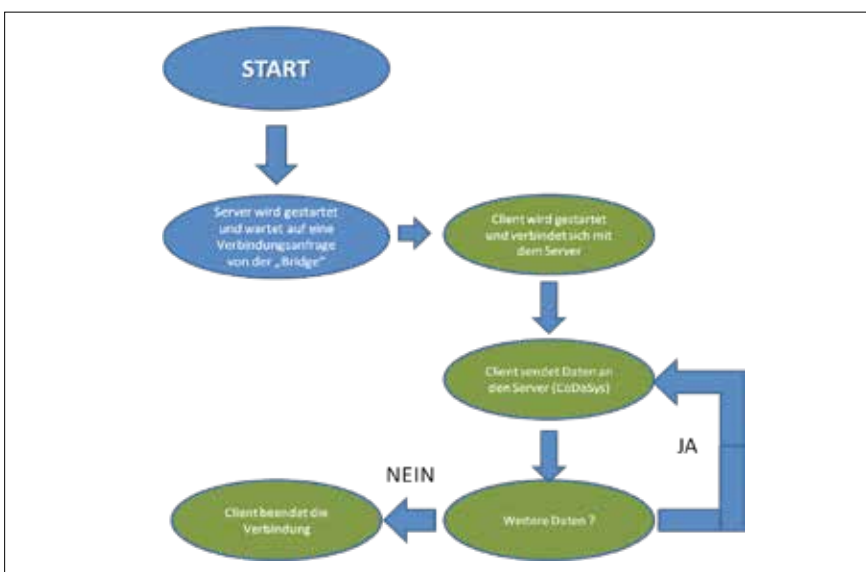


Abb. 3: Funktionsweise des Sockets.

der Spracherkennungssoftware hin zur Soft-SPS. Die Java-Bridge wurde mit Hilfe eines Software Development Kits (SDK) der Firma CLT in der Programmiersprache Java von uns entwickelt. Als Soft-SPS haben wir die CoDeSys der Firma 3S Smart-Software-Solutions verwendet, da diese Soft-SPS bereits häufig in der Industrie verwendet wird und als Demoversion kostenlos zu erhalten ist. Der letzte Teil des Systems ist die Hardwareverknüpfung. Über ein CANBus-Kabel wird der PC mit einer Input/Output-Karte verbunden, wo die Signale an die einzelnen Motoren der anzusteuernenden Geräte verteilt werden und anschließend ausgeführt werden.

An einem Beispiel kann man sich das wie folgt vorstellen (siehe Abb. 2): Ein Benutzer spricht den Befehl „Motor an“ in das Mikrofon des Headsets (1). Dieser Befehl gelangt via Kabel zu der Spracherkennungssoftware. Dort wird der Begriff mit den vorgegebenen Befehlen verglichen und eine „1“ (2) ausgegeben. Diese Zahl wird daraufhin von der Java-Bridge eingelesen und in ein array umgewandelt. Dabei wird sie neu codiert: 0100000000 (3). Dieser Code wird zu der Soft-SPS gesendet und dort überprüft, um welchen Befehl es sich handelt. Im gewählten Beispiel soll der Motor angeschaltet werden. Anschließend werden die Sicherungen im Programm überprüft: Das Programm muss im richtigen Modus laufen, der Motor muss gefahrlos aktiviert werden können und keine andere Sicherheitsvorkehrung darf den Befehl blockieren. Je nach Prüfergebnis wird der Befehl entweder freigegeben oder blockiert. Im Falle einer Freigabe aktiviert der Befehl via CANBus-Kabel den Motor (4).

### 3 Softwarespezifische Umsetzung

Der Datenfluss in unserem System sieht wie folgt aus: Der Sprachbefehl wird von einem Mikrofon aufgenommen, an die Spracherkennungssoftware gesendet und dort zugeordnet. Daraufhin liest die Java-Bridge die Daten der Spracherkennungssoftware aus.

Mithilfe einer TCP/IP Socketverbindung, einem Softwarelink zwischen den Programmen (siehe Abb. 3), wird die Verbindung zu der SPS hergestellt



und fortan werden Daten in einer Gruppe („array“) versendet. Nachdem der Socketserver (Empfänger) in der SPS das „array“ empfangen hat, wird dort der Wert der Variablen einer dementsprechenden Maschinenhandlung zugeordnet und diese ausgeführt.

Für die Socketverbindung benutzen wir folgende Bausteine:

- unseren selbstprogrammierten TCP/IP Java Client, der Bestandteil der Java-Bridge ist.
- den von uns modifizierten TCP/IP Server, ein Bestandteil der Soft-SPS, der auf der Programmieroberfläche der Soft-SPS hauptsächlich in der Programmiersprache „Structured Text“ programmiert wurde. Er dient der Verbindung der Java-Bridge zur Maschinensteuerung auf der Soft-SPS.

Unser TCP/IP Server und unser TCP/IP Client haben Ports, um Daten zu senden oder zu empfangen. Die Basis zur TCP/IP Programmierung in Java bildet das Package „Java.Net“. Aus der Klasse „Socket“ nutzen wir die Methode „getInputStream()“ zum Empfangen und „getOutputStream()“ zum Versenden von Daten. Die Ports zur Kommunikation haben eine Adresse, also eine Zahl, die natürlich bei Server und Client identisch sein muss. Die Java-Bridge sendet nach dem Sortie-

ren der Daten ein „array“ von 10 einzelnen Bytes vom Client zum Server (z. B. eine 1 als zweite Zahl, um einen Motor anzusteuern). Da der von der Spracherkennungssoftware importierte Variablentyp (.val) nicht in ein „array“ geschrieben werden kann, werden sämtliche Dateien zum Bearbeiten in den Typ „Bytes“ konvertiert. Diese lassen sich grundsätzlich einfacher bedienen und sind dementsprechend ideal.

Zum Aktivieren der Socketverbindung wird der TCP/IP Server gestartet, dann der TCP/IP Client. Nun sendet der Client Daten an den Server (in unserem Projekt sind das die Daten von der Spracherkennungssoftware). Falls der Client nun noch weitere Daten senden soll, wiederholt sich der Vorgang. Sobald der Client nichts mehr zu senden hat, wird die Verbindung beendet. Um eine sichere, permanente Verbindung zu haben, sendet der Client stets Nullen, damit die Verbindung nicht unterbrochen werden kann. Als Verbindung zwischen der Soft-SPS und den Input/ Output Karten haben wir eine CANBus-Kabelverbindung verwendet, da diese einfach und industrieller Standard ist. Dort werden die Signale vom PC aufgenommen und zu den entsprechenden Kabeln geleitet, die zu den Aktoren der Geräte führen. So sind beispielsweise die Motoren exakt steuerbar. In die andere Richtung werden hier die Signale der Sensoren gebündelt und durch das CANBus-Kabel an den PC gesandt.

## 4 Sicherheit

Je nach anzusteuernem Gerät sind in der Soft-SPS verschiedene Sicherheitsvorkehrungen zu programmieren, da jedes Gerät unterschiedliche Sensoren und Aktoren hat. Wenn bei unserem Beispielprojekt (siehe Kapitel 5) Hochregallager der Hebekran an der untersten Position ist, werden Sicherungen aktiviert, sodass man nicht mehr den Befehl „nach unten fahren“ aktivieren kann. Das Programm prüft permanent die aktiven Sensoren, um nur ordnungsgemäße Motorbewegungen zuzulassen.

Bei einer Sprachsteuerung ist grundsätzlich der unsicherste Bereich die analoge Wahrnehmung des Sprachbefehls, wobei mit schlechteren Mikrofonen die Fehlerkennungsrate steigt. Dennoch gibt es einige Punkte, mit der Sicherheit bei dem System deutlich erhöht werden könnte:

- Das Wiederholen eines Wortes in einer bestimmten Zeit: Die Fehlerquote ist bedeutend geringer, wenn man in einer bestimmten Zeit ein Wort mehrmals wiederholen muss. So haben wir in der Soft-SPS einen „Sicherheitsmodus“ einprogrammiert, bei dem ein Timer und ein Counter aktiviert werden. Nur wenn der entsprechende Befehl in einer bestimmten Zeit eine festgelegte Anzahl gesprochen worden ist, wird der Befehl ausge-

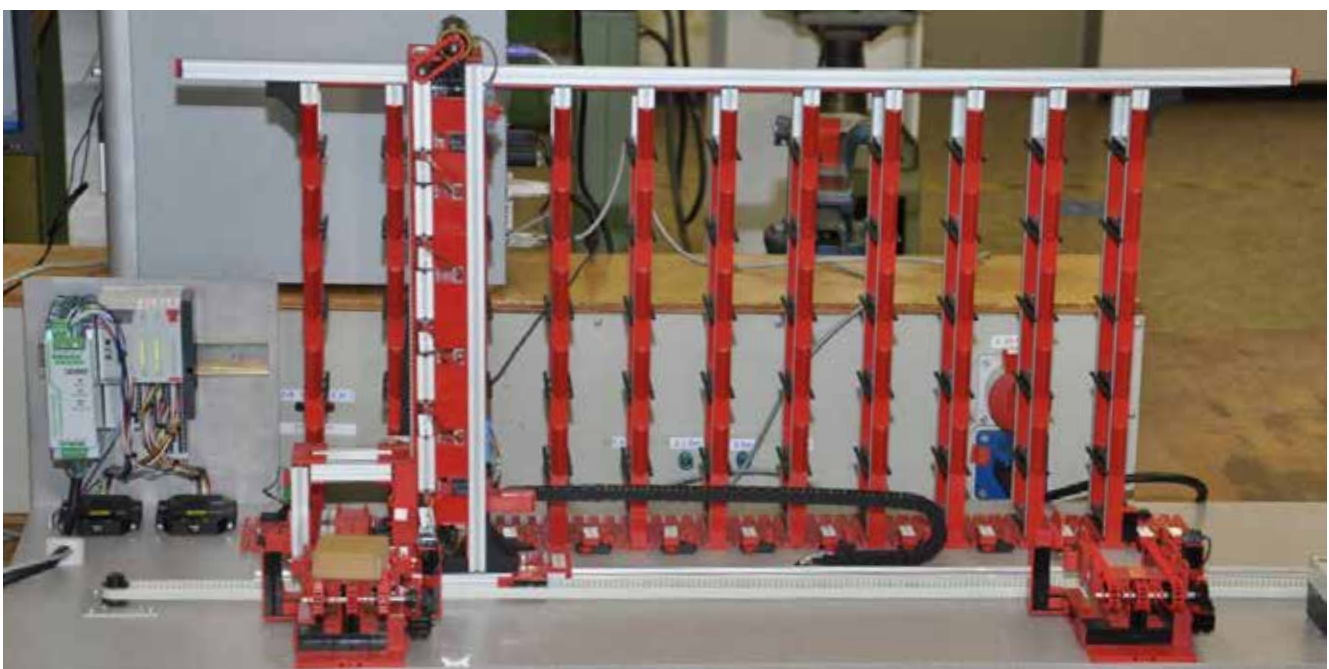


Abb. 4: Um das Gesamtsystem zu demonstrieren, wurde ein Hochregallager als Modell gebaut.

führt, ansonsten der Counter zurückgestellt. Dies ist bei besonders sicherheitsrelevanten Befehlen zu empfehlen.

- Ein weiteres Eingabemedium zur Aktivierung der Spracherkennung. Die Spracherkennung z. B. mit einem Knopf zu aktivieren (Muteknopf) ist die sicherste Methode falsche Befehle auszufiltern, wenn man selber nicht spricht. Auch eine Bewegungserkennung oder dergleichen wäre anwendbar, um immer noch ein bestätigendes Signal zu haben.

Zwar sind bei einem analogen Medium wie der Sprache Fehler nicht komplett auszuschließen, doch mit solchen Sicherheitsmethoden können sie auf ein Minimum reduziert werden. Zudem sollte aber jedes per Sprache zu bedienendes Gerät einen Notstop-Schalter oder eine Sicherung besitzen.

### 5 Beispiel Hochregallager

Um unser System bestmöglich demonstrieren zu können, haben wir unser System an einem Hochregallagermodell (siehe Abb. 4) und an drei Haushaltsgeräten (Radio, Rundumleuchte und Ventilator) umgesetzt. Das Hochregallagermodell besitzt fünf Motoren, die alle einzeln angesteuert werden können, und 26 Sensoren, auf die unser System reagieren kann. Die Sensoren sind mit einer 24 Volt Eingangskarte und die Aktoren mit einer 24 Volt Ausgangskarte verbunden. Diese beiden I/O Karten tauschen die nötigen Daten mittels CANbus Koppler mit dem PC aus. Als Gegenstelle wird im PC eine CANBus Einsteckkarte verwendet. Zudem wurden einige Ausgänge benutzt, um Haushaltsgeräte anzu steuern zu können. Das 24 Volt Signal der Ausgangskarte wurde verwendet, um ein Koppelrelais zu schalten, mit dem dann die 230 Volt Steckdosen aktiviert, bzw. deaktiviert werden können.

Mit diesen Verbindungen sind wir in der Lage, bei dem Hochregallager jeden einzelnen Motor per Sprache zu steuern oder einprogrammierte Abläufe zu starten. So können wir mit einem Sprachbefehl einen Gegenstand in ein Fach ein- oder auslagern. Damit können wir demonstrieren, dass sich auch mühelos komplexere Anlagen steuern lassen, indem wir jeden Motor genau kontrollieren und unser Programm zudem auf alle Sensoren achtet. Um zu zeigen, dass wir verschie-

dene Geräte auch parallel bedienen können, haben wir uns dafür entschieden, die Haushaltgeräte mit einem Koppelrelais zu aktivieren bzw. zu deaktivieren (theoretisch könnten wir aber auch diese komplett kontrollieren).

### 6 Anwendbarkeit

Die von uns entwickelte Verbindung, die Java-Bridge, zwischen einer Spracherkennungssoftware und einer kommerziellen Robotersteuerung ist nach unserem Kenntnisstand bisher einmalig. Ein großer Vorteil dieses Systems ist es, dass man es universell einsetzen kann. Da die SPS prinzipiell mit beliebigen Geräten verbunden werden kann, erreichen wir durch unsere Verbindung für die Sprachsteuerung eine große Schnittstelle. Je nach Anwendung muss lediglich eine neue Soft-SPS Applikation geschrieben werden, da jedes mechanische System andere Sensoren und Aktoren verwendet. Zudem sind auch die internen Maschinenabläufe unterschiedlich. Die von uns entwickelte Ankopplung der Spracherkennungssoftware bleibt allerdings in allen Fällen gleich. Der Kern des Systems bis zur SPS ist immer identisch und muss nicht verändert werden.

Durch die Universalität der Sprachsteuerung bestehen fast keine Grenzen in der Anwendbarkeit. Da sie mit einer SPS verbunden ist, lassen sich praktisch alle Geräte, die mit Strom funktionieren oder bei denen man beispielsweise Motoren anbringen könnte, so per Sprache steuern.

Einsetzbar wäre diese Sprachsteuerung zum Beispiel im Pflegebereich für körperlich benachteiligte Menschen. Hierbei könnte der Alltag deutlich erleichtert werden. Sie könnten zum Beispiel ihr Bett verstellen, ohne einen Pfleger herbeirufen zu müssen. Es könnten auch Radiosender verstellt oder Lichter an- beziehungsweise ausgeschaltet werden. Beides würde die Arbeit einer Pflegekraft erleichtern und den Patienten ein Stück mehr Lebensqualität geben.

Auch der Alltag von nicht körperlich beeinträchtigten Menschen könnte mit der Sprachsteuerung komfortabler werden. Ein Beispiel hierfür wäre das automatische Zubereiten von Kaffee.

Wenn man diese Sprachsteuerung im Haushalt verwendet, könnte man das System skalieren. So wäre es möglich einen kompletten Haushalt zu vernetzen,

indem auf nur einem PC die Sprachsteuerung läuft und alle weiteren Geräte mit diesem via Netzwerktechnik, wie LAN oder WLAN, verbunden sind. Zusätzlich könnte man die Socketverbindung über Internet laufen lassen, was den Vorteil mit sich brächte, dass einerseits der Server mit der Sprachsteuerung nicht im Haushalt stehen, sondern auf ein externes Zentrum ausgelagert werden würde. Andererseits ist es damit möglich, Geräte zu bedienen, ohne dass eine Person im Haus sein muss.

Die Sprachsteuerung wäre aber auch in der Industrie anwendbar: So könnten z. B. in der chemischen Industrie Verfahrensschritte per Sprachbefehl als Bedieneingabe gestartet werden. Auch kann durch eine Sprachsteuerung ein übergroßes Bedienfeld ersetzt werden und man könnte mit der Sprache etwas betätigen, auch wenn beide Arme bereits im Einsatz sind.

Zusätzlich besteht die Möglichkeit, dass neben oder anstatt der Spracherkennung Gestikererkennung verwendet werden könnte, indem man die Anbindung an unsere Socketverbindung universeller gestaltet. So müsste man auch nicht mehr eine bestimmte Spracherkennungssoftware verwenden.

### Danksagung

Zunächst möchten wir uns bei Knut Langhans für die Betreuung der Arbeit bedanken. Wir wollen auch der Firma DOW Chemical für das Sponsoring danken. Ebenfalls bedanken wir uns bei Gerd von Borstel, der uns in die Benutzung der Ausbildungswerkstatt der DOW Stade eingewiesen hat. Auch den drei Auszubildenden der Firma DOW (Eike Dralle, Jannek Hoops und Lennart Struve) sind wir für die Hilfe bei der Hardware dankbar. Des Weiteren bedanken wir uns bei der Firma Peak für die CANBus Karte, der Firma EATON für das I/O Modul und bei der Firma 3S für erforderliche CoDeDys-Lizenzen. Zusätzlich bedanken wir uns bei Jan Teegen vom Java-Club Elmshorn für einen Java-Crashkurs und bei der Firma Reinholz für wertvolle Informationen zur Programmierung. Wir möchten uns auch bei Annelie Menck und Stephanie Hilck für die Erstellung unserer Homepage ([www.vincent-voice-control.de](http://www.vincent-voice-control.de)) bedanken. Zuletzt bedanken wir uns bei Sven Heeschen für die Erstellung einer Patentschrift und bei unseren Eltern.