

Dezentrale Intelligenz

Jugend forscht 2006

Mathematik-Informatik

vorgelegt von

Stefan Krumpen

und Giugliano Wutschka

St. Michael-Gymnasium Jgst. 12



Inhaltsverzeichnis

1. Kurzfassung	2
2. Einleitung und Zielsetzung	3
3. RFID-Chips	4
3.1 Was sind Transponder und wie funktionieren sie?.....	4
3.2 Wie können Transponder als „dumme“ Einzelteile eine Dezentrale Intelligenz bilden?	4
4. Stand der Forschung	4
5. Unsere ersten Programme	5
6. Ideen für die Umsetzung in die RFID Praxis	7
6.1 Die dezentral verwaltete Lagerhalle	7
6.2 Der virtuelle Ameisenstaat	11
7. Kritische Reflexion und Ausblick	13
8. Links.....	13



1. Kurzfassung

Dezentrale Intelligenz ist eine Form von Intelligenz, bei der keine zentrale Steuerung vorliegt. Das beste Beispiel hierfür ist unser Gehirn: Hier arbeiten Milliarden von Nervenzellen intelligent zusammen, ohne dass eine übergeordnete Instanz ihnen sagen muss, was sie tun sollen. Solche dezentrale Steuerungsprozesse sind bei komplexen Systemen zentralen Steuerungen weit überlegen, da sie kreativ auf sich ändernde Sachverhalte reagieren können. Ein weiteres Beispiel für eine Dezentrale Intelligenz ist z.B. ein Ameisenstaat.

Auf den oben aufgeführten Gedanken beruht auch unsere Jugend-forscht-Idee:

Warum sollen zahlreiche „dumme“ RFID-Chips - die miteinander in Wechselwirkung stehen - nicht auch eine Dezentrale Intelligenz ausbilden können, die Probleme kreativ lösen kann und die somit den bisherigen zentral gesteuerten Ordnungssystemen überlegen ist.

In unserer Arbeit haben wir ein Simulationsprogramm für ein sich selbst organisierendes Lager geschrieben, in dem sich mit RFID-Chips bestückte Waren von selbst in Regale einordnen. Auch haben wir eine Architektur für einen solchen Chip angedacht. Des Weiteren haben wir einen Ameisenstaat im Computer simuliert, bei dem sich die Ameisen genau wie in der Natur organisieren und Ordnung schaffen.



2. Einleitung und Zielsetzung

Die Vorfahren des Menschen und auch die Menschen selbst lebten bis vor wenigen Jahrhunderten in einer relativ einfachen und überschaubaren Welt. Aus diesem Grund hat sich im Rahmen der Evolution unser Gehirn so entwickelt, dass es die zum Überleben notwendigen linearen Denkprozesse gut beherrscht (siehe [1]). Da unsere heutige Welt aufgrund der weltweiten Vernetzung und des rasant wachsenden Wissens immer komplexer wird, stoßen lineare Denkprozesse und zentrale Steuerungsmechanismen aber bei komplexen Systemen sehr schnell an ihre Grenzen. Beispiele für solch komplexe Systeme sind wirtschaftliche und politische Systeme. Wie man im Geschichtsunterricht lernt, entziehen sich komplexe Systeme mit der Zeit einer linear dirigistischen Lenkung und neigen zur Selbstorganisation. Weitere Beispiele für kreative Selbstorganisation finden sich in der Chaostheorie und in der Natur. Ein Beispiel sind die äußerst komplexen und trotzdem stabilen Biotope. Eines der komplexesten natürlichen Systeme ist aber unser Gehirn. Obwohl es hier keinen zentralen Herrscher gibt, organisieren sich die „dummen“ Nervenzellen zu einer sinnvollen Einheit, die auf neue Sachverhalte kreativ reagieren kann. Dieser Prozess der Selbstorganisation, der auch „Dezentrale Intelligenz“ genannt wird, hat sich bei komplexen Systemen in der Natur als äußerst sinnvoll erwiesen, da er die, bei sich verändernder Umwelt auftretenden Probleme, kreativ lösen kann. Auf den oben aufgeführten Gedanken beruht auch unsere Jugend-forscht-Idee:

Warum sollen zahlreiche „dumme“ RFID-Chips - die miteinander in Wechselwirkung stehen - nicht auch eine Dezentrale Intelligenz ausbilden können, die Probleme kreativ lösen kann und die somit den bisherigen zentral gesteuerten Ordnungssystemen überlegen ist.

In unserer Arbeit wollen wir diese Chips aber nicht bauen, da uns hierzu die technischen Möglichkeiten fehlen, sondern wir wollen eine Architektur hierfür entwickeln und diese im Computer auf ihre Funktion hin überprüfen. Wir glauben damit einen Beitrag zu leisten, dass Menschen in einer komplexen Welt nicht durch dumme, zentrale Steuerungen drangsaliert werden.



3. RFID-Chips

3.1 Was sind Transponder und wie funktionieren sie?

Radio Frequency Identification ist eine Methode, mit der man Daten berührungslos lesen und speichern kann. Ein RFID-System besteht aus einem Transponder und einer Sende- und Empfangseinheit. Die Transponder werden mittels Radiowellen beschrieben bzw. ausgelesen. Die Energieversorgung des Transponders funktioniert mittels Induktion über Radiowellen (passiv) oder über eine im Transponder eingebaute Batterie (aktiv). Die Reichweite des Systems variiert je nachdem, ob der Transponder seine Informationen aktiv sendet, oder seine Energie über die Radiowellen der Sendereinheit erhält.

3.2 Wie können Transponder als „dumme“ Einzelteile eine Dezentrale Intelligenz bilden?

Ein Gehirn (Dezentrale Intelligenz) besteht aus vielen „dummen“ Nervenzellen. Auch Transponder könnten unserer Meinung nach eine Dezentrale Intelligenz bilden, wenn auf jedem von ihnen ein kleines Programm gespeichert ist und sie mit benachbarten Transpondern in Wechselwirkung treten können.

Nimmt man dagegen einen Ameisenstaat als Beispiel für eine Dezentrale Intelligenz, so müssen sich die Transponder fortbewegen können oder bewegt werden. Wenn jetzt sehr viele Transponder zusammen kommen und sich gegenseitig austauschen, könnten sie in der Lage sein ein intelligentes Verhalten zu entwickeln.

4. Stand der Forschung

Um herauszufinden, ob wir mit unserer Arbeit auch wirklich etwas Neues entwickeln, suchten wir im Internet, ob es schon eine Anwendung gibt, bei der Transponder eine Dezentrale Intelligenz bilden oder zumindest dezentral zusammenarbeiten. Obwohl das Wort „Dezentrale Intelligenz“ häufig verwendet wird, fanden wir jedoch nur Quellen, die sich mit dem bisherigen Einsatz von RFID beschäftigen (Logistik Branche, Warenidentifikation,...). Dezentrale Systeme, die kreativ durch Selbstorganisation auf neue Umweltbedingungen



reagieren können, fanden wir nicht. Diese Eigenschaft ist jedoch der Hauptvorteil der Dezentralen Intelligenz.

Wir glauben also mit unserer Arbeit über Dezentrale Intelligenz mit Transpondern Neuland zu betreten.

5. Unsere ersten Programme

Unser erstes Problem bestand darin, dass man ganz viele Transponder braucht, um eine Dezentrale Intelligenz zu erschaffen. Da wir uns aber keine tausend Transponder kaufen können, haben wir die Transponder in Computer simuliert.

Jedes unserer Programme schrieben wir zuerst in Quickbasic, weil das eine sehr einfache Programmiersprache ist. Ein Nachteil von Quickbasic ist, dass es auf DOS-Ebene arbeitet und deshalb die Grafik (insbesondere die Bildschirmauflösung) nicht besonders gut ist. Aus diesem Grund haben wir jedes Programm das in QuickBasic funktionierte, zusätzlich in Delphi geschrieben. Das ist zwar etwas schwieriger, hat aber den Vorteil, dass das Ganze sehr viel besser aussieht, weil Delphi die Windowsoberfläche verwendet.

Die Transponder haben wir in Quickbasic als einzelne Pixel dargestellt, in Delphi zuerst als Label, später auch als Pixel oder Bild.

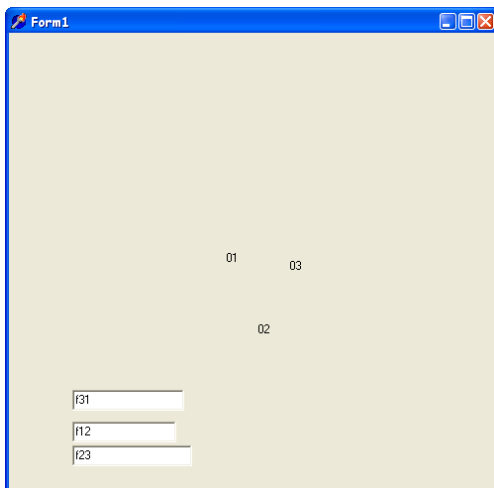


Abb.5.1: Unser erstes Programm, in dem sich drei Transponder(Labels) aneinander orientieren.

In unseren ersten Programmen haben wir zunächst nur drei Transponder simuliert, die sich aneinander orientieren. Sie bewegen sich nach dem Programmstart zufällig auf der Oberfläche und berechnen nach jedem Schritt die Entfernung zueinander mittels des Satzes von Pythagoras. Wenn die Entfernung kleiner als hundert Pixel ist, nimmt der eine Transponder die Richtung des anderen an.

Nach einer kurzen Zeit haben sich alle drei Transponder gefunden und „laufen“ synchron umher. Die erste Erweiterung war, dass die Transponder nicht mehr die Richtung des anderen übernehmen, sondern immer, nachdem sie einen anderen gefunden haben,

dem Abstand von hundert Pixeln beibehalten. Dasselbe haben wir dann für bis zu 100 Transponder simuliert.



Das war schon mal ein Erfolg für uns. Wenn aber viele Transponder von einem Programm simuliert werden, ist das nicht so richtig dezentral. Deshalb haben wir eine Methode entwickelt, mit der wir endlich viele Programme schreiben können, von denen jedes einen Transponder repräsentiert und die alle unabhängig voneinander sind, aber untereinander kommunizieren können. Diese Kommunikation verläuft über die Ebene von Textdokumenten, in welche die Programm-Transponder z.B. ihre Koordinaten schreiben und die der anderen Transponder einlesen. Das funktioniert aber nur in Delphi, da in Quickbasic nie mehr als ein Quickbasic-Programm gleichzeitig ausgeführt werden kann. Ein weiterer Vorteil dieser Methode besteht darin, dass die Transponder unterschiedliche Eigenschaften haben können, was bei einem zentralen Programm sehr aufwendig zu programmieren ist. Eine Weiterentwicklung war, dass die Transponder, da sie jetzt unterschiedliche Eigenschaften haben, unterschiedliche Aufgaben erfüllen können. In einer Demo haben wir drei Transponder durch drei unterschiedliche Programme geschaffen, von denen einer die Umgebung erkundet,

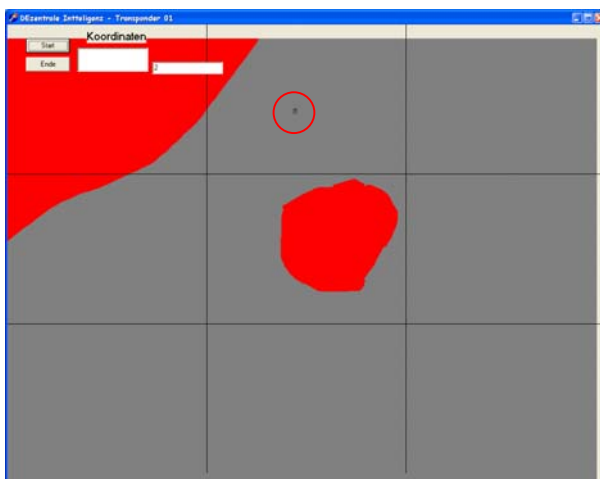


Abb.5.2: Unser Erkunderprogramm: der Erkunder (roter Kreis) bewegt sich zufällig auf der Oberfläche und macht rote Bereiche ausfindig.

welche aus einem Bild besteht und bei jedem Programm im Hintergrund ist. Auf diesem Bild befindet sich ein roter Fleck. Wenn der Erkunder diesen Fleck findet. Sagt er den anderen Transpondern wo der Fleck zu finden ist. Einer der anderen fühlt sich dann von diesen Fleck angezogen und bewegt sich in dessen Richtung. Der dritte Transponder bewegt sich von dem Fleck weg. Somit hatten wir eine Grundlage geschaffen, auf der wir unsere Dezentrale Intelligenz aufbauen konnten.



6. Ideen für die Umsetzung in die RFID Praxis

6.1 Die dezentral verwaltete Lagerhalle

6.1.1 Die Idee

Die erste Idee für eine Anwendung unserer bisherigen Entwicklung ist eine dezentral verwaltete Lagerhalle. In dieser soll jede Ware mit einem Transponder versehen werden, auf dem ein kleines Programm gespeichert ist. Kommt die Ware, beispielsweise eine Ketchupflasche, jetzt in das Lager, bewegt sie sich mit einem bestimmten Algorithmus durch das Lager und findet so automatisch ihren Platz. Während sie sich durch das Lager bewegt, guckt sie, welche Ware an der jeweiligen Stelle ihren Platz hat, und gibt diese Informationen für alle Waren frei. Wenn sie zum Beispiel an dem Regal für Colaflaschen vorbeikommt, sagt sie allen Colaflaschen wo diese hin müssen und die Colaflaschen gehen den direkten Weg zu ihren Platz.

Ein solches Lager haben wir für 100 Waren (15 verschiedene Arten) simuliert: In ein beliebig aufgebautes Lager eingebrachte Waren bewegen sich auf einer nach innen kleiner werdenden Spirale durch das Lager. Wenn sie auf einen Platz treffen, der für sie vorgesehen ist, besetzen sie diesen. Zusätzlich sagen sie den anderen Waren auf welche Plätze sie auf Ihrem Weg gestoßen sind. Die anderen bewegen sich dann direkt zu diesen Plätzen hin, wenn diese Plätze für sie vorgesehen sind. Die verschiedenen Waren haben wir als Quadrate mit verschiedenen Farben dargestellt, die Plätze als Winkel mit verschiedenen Farben. Wenn eine Ware einen freien Platz in ihrer Farbe gefunden hat und diesen besetzt, füllt sie diesen zu einem großen Quadrat aus. Des Weiteren kann man die Ware auf Wunsch wieder aus dem Lager entsenden, wobei die mit dem ältesten Verfallsdatum zuerst entsendet werden.

Diese dezentrale Lagerverwaltung reagiert ohne Zutun eines Menschen durch Selbstorganisation kreativ auf sich ändernde Lagerbedingungen. Das System ist eine „Dezentrale Intelligenz“. Wenn jetzt zum Beispiel das Regal für die Ketchupflasche an eine andere Stelle versetzt wird, reagieren die Ketchupflaschen entsprechend und gehen den neuen Weg.

Da die Ware sich nicht selber fortzubewegen kann, muss es ein System von Fließbändern in der Lagerhalle geben, über das die Ware an den jeweiligen Platz gelangt; oder Roboter bringen die Ware an Ihren Platz. In beiden sagt aber die Ware wo sie hin will.

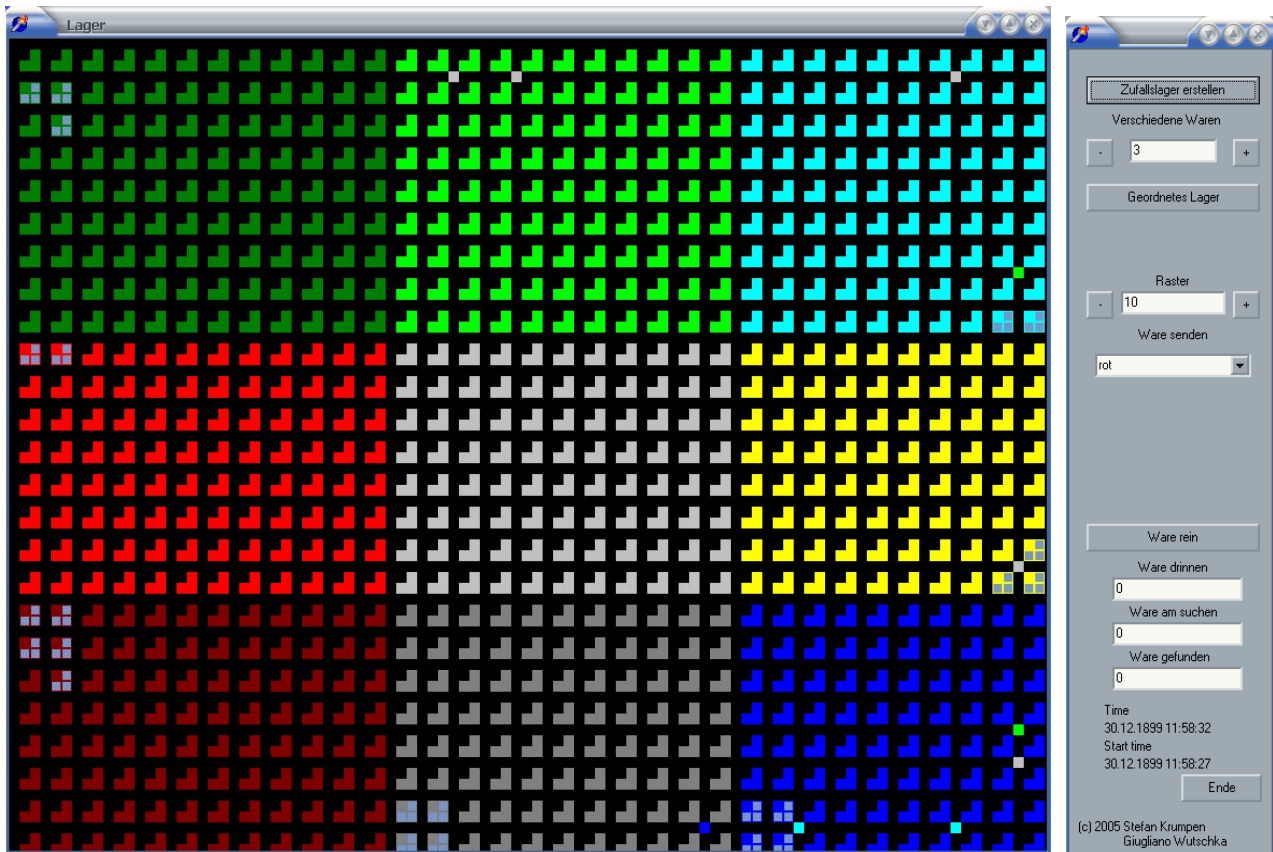


Abb. 6.1: Unser Lagersimulationsprogramm

Links: Eine Anordnung von Plätzen (Winkel) in der verschiedene Waren (kleine Quadrate) Plätze in ihrer Farbe suchen. Die großen Quadrate sind schon belegte Plätze.

Rechts: Die Steuerleiste des Programms.

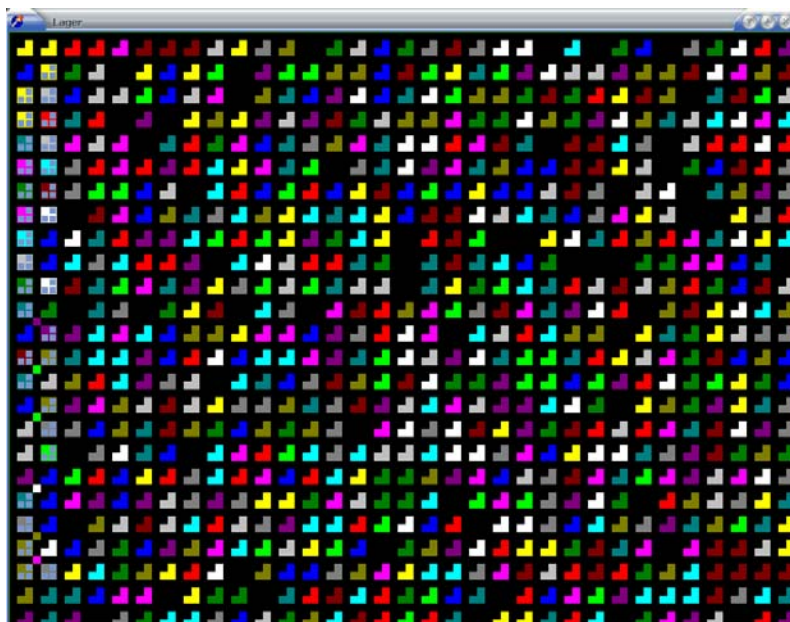


Abb.6.2: eine andere Anordnung von Plätzen: auch hier findet jede Ware ihren Platz.



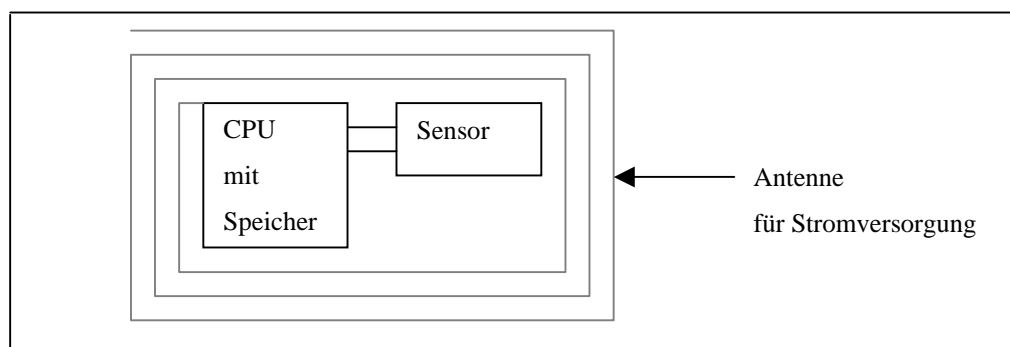
6.1.2 Planung eines Chips für unser Programm

Ein RFID-Chip, für unser Lagerprogramm, muss einen Speicher für das Programm, sowie eine kleine CPU und einen Sensor besitzen der verschiedene Farben erkennen kann. Seine Energie bezieht er über Radiowellen, die von einem im Lager angebrachten Sender gesendet werden. Außerdem muss er mit dem Transportsystem kommunizieren können, um diesem sein Ziel angeben zu können. Auch muss er mit anderen sich im Lager befindlichen RFID-Chips bzw. Waren in Verbindung treten können, um diesen ihren Bestimmungsort anzugeben.

Die CPU muss also in der Lage sein, folgende kleine Rechenoperationen durchzuführen:

- 1.: Die Art der umgebenden Plätze über den Sensor zu erkennen und den Weg zu diesen berechnen.
- 2.: Dem Transportsystem sagen wohin es die Ware bringen soll und anderen Waren sagen welche Plätze wo sind.

Der Schaltplan für unseren Chip könnte so aussehen:

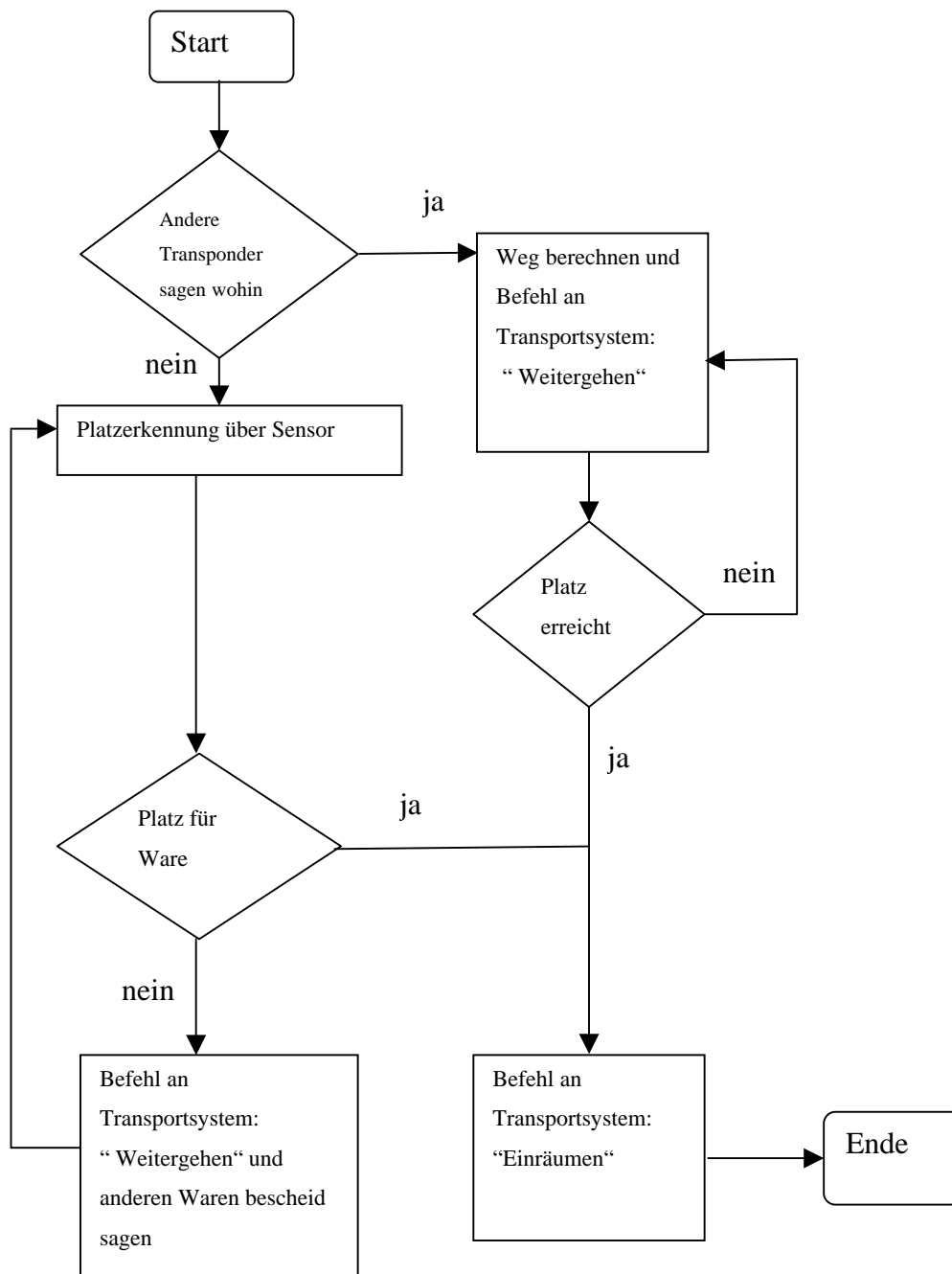


Die Größe des Speichers für das Programm muss zwischen 100 und 300 kb liegen, für die CPU sollte eine Geschwindigkeit von 15MHz ausreichend sein. Weiterhin muss die CPU über einen kleinen Sender verfügen um mit anderen Transpondern und dem Transportsystem in Kontakt treten zu können. Der Sensor muss in der Lage sein, verschiedene Farben über Entfernungen bis mehrere Meter (hängt vom Lager ab) präzise zu erkennen. Ein Beispiel für die Funktionsweise eines solchen Sensors ist, dass weißes Licht gesendet wird und das reflektierte Licht in seine Grundfarben zerlegt und aus diesen die Farbe des Platzes erkannt wird. Bei alledem muss beachtet werden, dass der Preis für einen Chip in einem gesunden



Verhältnis zum Nutzen steht. Die Entwicklung im Bereich der organischen Elektronik lässt hoffen, dass die Preise für RFID-Chips in Zukunft deutlich sinken. Es muss auch nicht jede einzelne Ware einen Transponder besitzen. Es reicht, wenn zum Beispiel an jeder Palette ein Chip befestigt ist.

Die Struktur des Programms sieht folgendermaßen aus:





6.2 Der virtuelle Ameisenstaat

Ein anderes Beispiel für eine Dezentrale Intelligenz stellt eine Ameisenkolonie dar. Auch hier gibt es keine zentral steuernde Instanz und keine Ameise überblickt das ganze System, jedoch ergibt das Zusammenwirken aller ein sinnvolles Ganzes. Wir haben dieses Beispiel aufgegriffen und eine virtuelle Ameisenkolonie geschaffen. Die Kommunikation unter den Ameisen in der Natur verläuft unter anderem über Gerüche und Duftspuren. Findet z.B. eine Ameise irgendwo Futter, so legt sie eine Duftspur zurück zum Ameisenhaufen, die andere Ameisen gezielt zum Futter führt. Um die Gerüche zu simulieren haben wir die „Welt“ in der unsere Ameisen leben in Felder aufgeteilt. So kann jedem Feld eine Geruchsintensität zugewiesen werden. Unsere virtuellen Ameisen, die durch rote Pixel dargestellt werden, starten alle im Ameisenhaufen und laufen von da aus in verschiedene Richtungen und ändern ihre Richtung nach jedem Schritt. Stößt eine Ameise auf Futter (blaue Pixel), so nimmt sie ein Stück Futter und bringt es auf direktem Weg zurück zum Ameisenhaufen und legt dabei eine intensive Geruchspur. Andere Ameisen die diesen Geruch wahrnehmen folgen ihm. Nach jedem Schritt ändern sie ihre Richtung unter Einbeziehung des Geruchs der umgebenden Felder, wobei das Feld mit dem intensivsten Geruch bevorzugt wird, sodass sie zum Futter gelangen. Mit der Zeit entstehen richtige Ameisenstraßen, da immer mehr Ameisen dem Geruch der Anderen folgen. Wie in der Natur kennt keine unserer Ameisen die Position der anderen, jedoch arbeiten alle sinnvoll zusammen und bringen nach und nach alles Futter nach Hause.

Der Unterschied zu unserem Lager besteht darin, dass hier die Objekte (Ameisen) selber eine sinnvolle Ordnung schaffen (Futter sammeln und lagern). Anwendungen für das Ameisenprinzip finden sich überall dort, wo Ordnung geschaffen werden muss. Dagegen ordnen sich bei unserem Lager die Objekte in eine vorgegebene Ordnung ein.

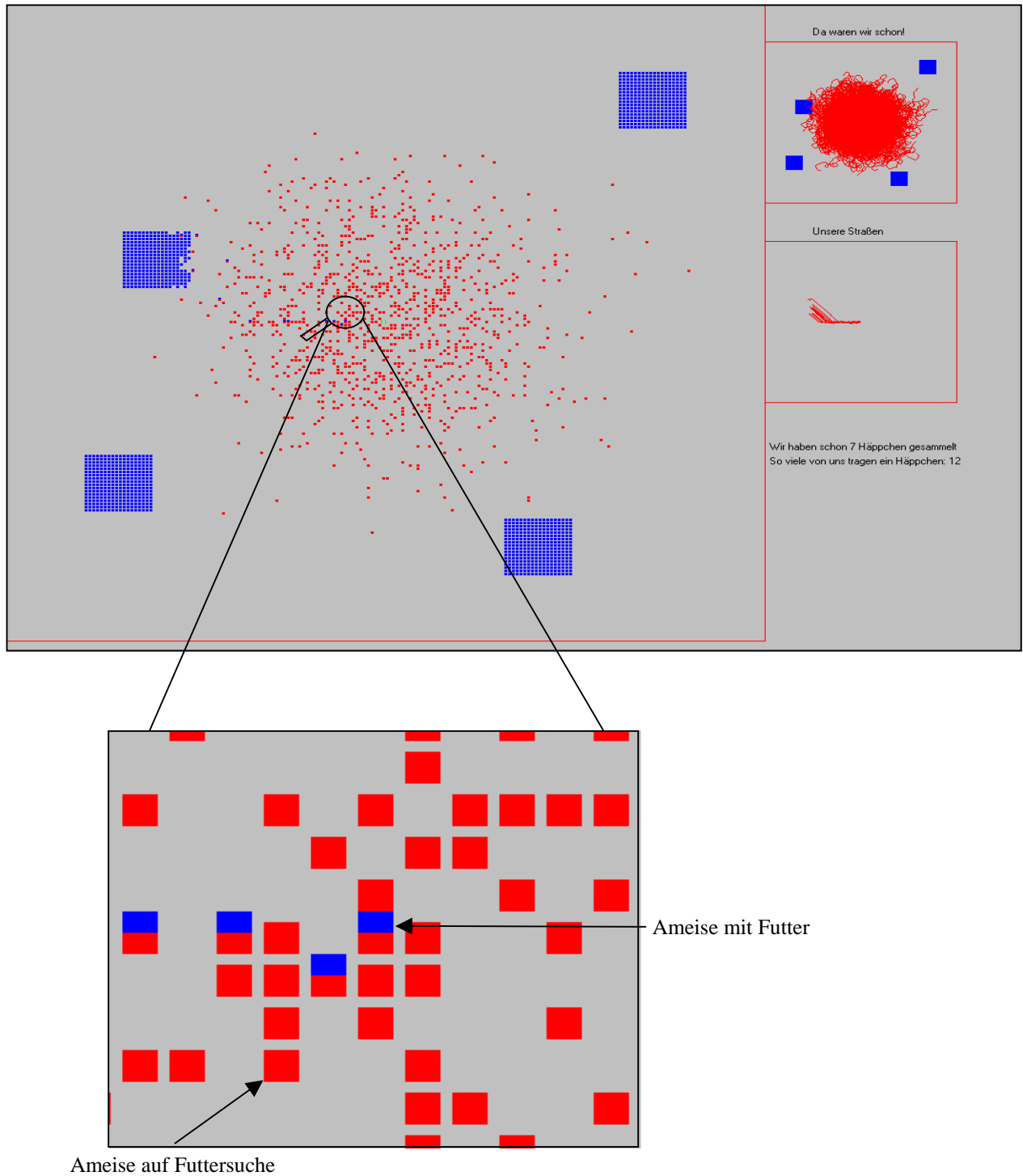


Abb.6.3 Unser Ameisenstaat:

Links: Die roten Punkte sind Ameisen auf Futtersuche, die mit einem blauen Punkt über sich tragen Futter. Die einzelnen blauen Punkte sind Futterhäppchen.
 Rechtsoben: Anzeige wo schon Ameisen hergelaufen sind.
 Rechtsmitte: Darstellung der entstandenen Ameisenstraßen.



7. Kritische Reflexion und Ausblick

Nach vielen Stunden des Programmierens, haben wir zwei Programme geschaffen, welche auf sich ändernde Sachverhalte kreativ reagieren können. Außerdem haben wir eine Architektur für einen RFID-Chip angedacht, der den Bau von sich selbst organisierenden Lagern ermöglicht. Bevor wir diesen Plan weiterentwickeln, wollen wir uns mit einem Experten für Lagerverwaltung bei der DHL unterhalten, da er über die nötige praktische Erfahrung verfügt.

Bis zum Wettbewerb werden wir unser Ameisenprogramm erweitern und nach neuen Ideen für eine Dezentrale Intelligenz in RFID-Netzen Ausschau halten.

8. Links

- [1] http://www.mpg.de/bilderBerichteDokumente/multimedial/mpForschung/2005/heft02/2_05MPF_14_18.pdf

- [2] http://www.m-lab.ch/rfid-workshop/hannover_paper.pdf

- [3] http://de.wikipedia.org/wiki/Neuronale_Netze

- [4] <http://de.wikipedia.org/wiki/RFID>