

Demand Side Management durch intelligente Software-Agenten und Feldbus Technologie

Peter Palensky

Institut für Computertechnik, TU Wien
Gußhausstraße 27-29, A-1040 Wien
{palensky@ict.tuwien.ac.at}

1 EINLEITUNG

Die Deregulierung und Liberalisierung des europäischen Strommarktes bietet eine Fülle an Chancen und Risiken. Der Konkurrenzdruck für EVU wird stärker, der Wettbewerb intensiviert. Diese Veränderungen werden die EVU in zwei Gruppen teilen: in die Gewinner und die Verlierer. Neben vielzähligen wirtschaftlichen Vorbereitungen wie Kooperationen und Fusionen mit starken internationalen Partnern gibt es eine weitere Karte die ausgespielt werden kann und wird. Innovation.

Technischer Fortschritt am Energiesektor ist in Zukunft unabdingbares Element eines konkurrenzfähigen EVU. Man hebt sich dadurch nicht nur von Mitbewerbern ab sondern kann seine eigene Erzeugung und Verteilung kostengünstiger gestalten. Letztendlich wird die Entscheidung in großem Ausmaß vom Preis des Produktes geprägt. Daher muß man ständig Wege finden, die es ermöglichen, die Versorgung der Kunden mit einem Minimum an Kosten bereitzustellen.

Dieser Beitrag handelt von einem neuen Ansatz den Energieverbrauch zu koordinieren und zu steuern.

2 DEMAND-SIDE-MANAGEMENT

Demand-Side-Management (DSM) ist der Überbegriff für verbraucherseitige Maßnahmen des Energiemanagements. Im Speziellen ist aber fast immer Lastgangsglättung gemeint. Man versucht den Gesamtverbrauch einer Anzahl von Verbrauchern konstant zu halten. Jedes System energiekonsumierender Geräte hat einen charakteristischen Lastgang. Ein privater Haushalt hat eine „Mittagsspitze“, ein industrieller Betrieb andere typische Spitzenlasten.

Solche Spitzenlasten kann man durch die Verwendung von gespeicherter Reserveenergie abdecken. Pumpspeicherkraftwerke werden zum Beispiel für solche Zwecke eingesetzt. Aber auch andere Energieformen können als Speicher verwendet werden. Ein Nachteil solcher Lösungen ist, daß Energiespeicher im allgemeinen teuer sind. Nur in Extremfällen rechnet sich die Spitzenlastabdeckung durch Energiespeicher.

Wesentlich sinnvoller ist es, das Problem an der Wurzel zu packen. Wenn man den Lastgang an der Verbraucherseite glätten kann, ist die Notwendigkeit von Energiespeichern nicht mehr gegeben. Das erreicht man üblicherweise durch zeitliche Koordination der Verbraucher. Anstatt alle Verbraucher gleichzeitig Energie konsumieren zu lassen, teilt man sie so ein, daß sie es möglichst nacheinander tun.

In vielen Fällen ist das ein Problem. Auf den ersten Blick scheinen die System oft nicht dazu geeignet zu sein, die Verbraucher nacheinander arbeiten zu lassen, bei genauerer Betrachtungsweise kann man aber oft eine Vielzahl an Möglichkeiten erkennen, die Reihenfolge der Aktionen der Verbraucher sinnvoll zu beeinflussen.

Aber nicht nur die Problematik der Spitzenlasten und der dadurch verursachten Probleme in der Energieerzeugung und im Energietransport sind eine Motivation für DSM-Maßnahmen. Das EVU will sein Energieverteilungsnetz (Grid) auch für „Transits“, d. h. temporären Transport von Energie verwenden. Im deregulierten Markt wird es vermehrt dazu kommen, daß große Mengen an Energie von einem Ort zu einem anderen verschoben d. h. verkauft werden müssen. In einem solchen Fall kann es leicht passieren, daß das Grid kurzzeitig überlastet ist, wenn man seine Grundlast (d. h. den Energieverbrauch der „Standardkunden“) nicht kurzzeitig senken kann. Mit DSM könnte man auch aus diesem Grund eingreifen. Neben Transits sind auch andere Überlastsituationen denkbar, bei denen es wünschenswert wäre, das Netz bewußt kurzzeitig zu entlasten.

Herkömmliche DSM-Systeme teilen die Geräte nach Prioritäten ein, und schalten bei einer drohenden Überschreitung eines spezifizierten Maximalwertes zunächst die „unwichtigeren“ (d.h. niederprioren) Verbraucher ab, und, sofern das nicht reicht, später auch die höherprioren. Es gibt also eine mit den Verbrauchern vernetzte, zentrale Stelle, die sich der Prioritäten der Verbraucher bewußt ist, den aktuellen Gesamtenergieverbrauch mißt und die Entscheidungen trifft, wer eingeschaltet sein darf, und wer nicht. (Abbildung 1)

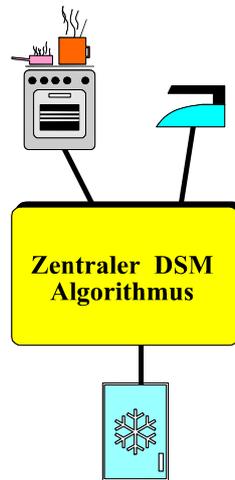


Abbildung 1: zentrales DSM

Diese Vorgehensweise ist für kleinere, starre Systeme brauchbar, weist aber bei komplexeren Systemen einige Nachteile auf:

1. *Komplexität*: Der Algorithmus der die Geräte einteilt, wird bei großen Systemen komplex, unübersichtlich, schwer wartbar und bedarf hoher (und damit teurer) Rechenleistung.
 2. *Unflexibilität*: Wenn ein Gerät dem System hinzugefügt wird, muß unter Umständen der Algorithmus geändert werden. Manche Algorithmen kann man parametrieren, andere müssen umprogrammiert werden.
 3. *Fehlertoleranz*: Wenn die zentrale Stelle ausfällt, so ist in diesem Fall das ganze System davon betroffen.
- Wenn man sich nun vorstellt, daß die besagten Verbraucher elektrische Geräte eines privaten Haushalts sind, offenbaren sich diese Nachteile ganz extrem.

Ad. 1: Die erhöhte Komplexität macht das System teuer und somit für den privaten Anwender unbrauchbar.

Ad. 2: Das System der Verbraucher in einem bewohnten Haushalt unterliegt keiner statischen Struktur. Geräte werden gekauft, andere wieder entfernt. Das System jedesmal umkonfigurieren zu müssen ist kaum möglich. Es ist im Falle eines professionellen Dienstes zu teuer und im Falle von do-it-yourself unzumutbar.

Ad. 3: Wie in anderen Anwendungen auch ist ein Ausfall des Systems hier nicht akzeptabel.

Was fehlt, ist ein neuer Ansatz für DSM, der diesen Anforderungen gerecht wird.

3 INTELLIGENTE SOFTWARE-AGENTEN

Das Forschungsgebiet „Verteilte Künstliche Intelligenz“ (VKI) behandelt unter anderem ein Teilgebiet, das „intelligente Software-Agenten“, „autonome Agenten“ oder kurz „Agenten“ genannt wird. Diese Agenten bieten eine Lösung der im vorigen Kapitel genannten Probleme an.

Ein Software-Agent ist im Prinzip nichts anderes als ein Programm. Er hat lediglich gewisse Eigenschaften, die ihn „intelligenter“ scheinen lassen und ihn somit von herkömmlichen Programm- und Softwaresystemen abheben. Zu definieren, was ein Agent ist, oder festzulegen, welche Eigenschaften ein Agent mindestens haben muß um ein Agent zu sein, ist seit Jahren einer der Hauptstreitpunkte der VKI ([Petrie97]). Es hat sich aber mittlerweile ein Grundtenor herauskristallisiert: Ein Agent ist ein Programm, das ein Ziel verfolgt und dieses durch eigene oder hinzugezogene Fähigkeiten autonom zu erreichen versucht. Objektorientiert betrachtend kann man festhalten, daß ein Agent ein Objekt ist, das einen Plan hat. Um diesen Plan zu verfolgen hat es gewisse Fähigkeiten. Mit der Hilfe von Sensoren, Aktoren und Kommunikationsschnittstellen kann das Objekt mit seiner Umgebung und mit anderen Agenten interagieren. Der Agent bedarf dabei keiner ständigen Bedienung wie ein „normales“ Programm sondern bekommt seine Aufgabe einmal mitgeteilt und agiert dann völlig selbständig. Wenn der Auftrag (die „Mission“) erfüllt ist, dann präsentiert der Agent die Ergebnisse an seinen Auftraggeber ([Brenner98]). Quantifizierend kann man sagen ein Agent ist um so mehr ein Agent, je mehr der folgenden Eigenschaften er aufweisen kann:

Eigenschaft	Bedeutung
Reaktivität	Reagiert auf Änderungen der Umgebung
Autonomie	Gehorcht nur seiner eigenen Natur
Zielorientiert/Proaktivität	Folgt einem Plan und initiiert selbständig Aktionen
Kontinuität	Ist ein kontinuierlich laufender Prozeß

Kommunikativ, sozial fähig	Kommuniziert und kooperiert mit anderen Agenten/Personen
Lernfähig	Benutzt seine Erfahrung
Mobilität	Kann selbsttätig von Maschine zu Maschine wandern
Flexibel	keine starre Ablaufsteuerung
Charakter	Wirkt in seinen Handlungen emotionell

Tabelle 1: Attribute für Agenten nach [Franklin97]

Ein Programm, das zum Beispiel auf Sensordaten reaktiv antwortet, kann also bereits als (sehr) einfacher Agent bezeichnet werden. Die Mobilität ist eine besondere Eigenschaft, die zu sogenannten „mobilen Agenten“ führt. Konkret bedeutet das, daß ein Agent von einem Computer, auf dem er gerade ausgeführt wird, beliebig auf einen anderen Computer wandern kann, um dort andere Agenten zu treffen, spezielle Informationen zu sammeln oder die individuellen Eigenschaften der Zielmaschine (z. B. besondere Input/Output-Fähigkeiten) zu nützen. Wenn man als Beispiel einen Agenten für die Steuerung der Raumtemperatur einsetzen will, wird dieser folgende Eigenschaften aufweisen:

- Reaktivität: Erkennt Temperaturschwankungen und bedient die Heizung entsprechend
- Autonomie: muß nach Wahl des Sollwertes nicht mehr bedient werden
- Ziel: Temperatur nach Wunsch des Benutzers konstant halten
- Kontinuität: führt seine Aufgabe ständig durch

Die restlichen Eigenschaften sind bei diesem Thermostat nicht vertreten. Er ist ein sehr einfacher Agent. Manchmal wird diese Funktionalität als nicht ausreichend angesehen, um einen Agenten zu bilden, und es wird gefordert, daß der Agent seine Meßdaten intelligent interpretiert ([Hayes95]).

Generell gilt, daß die Zielorientiertheit, die Autonomie und die Fähigkeit der Kommunikation als die wichtigsten Eigenschaften angesehen werden. Letztere kommt in sogenannten Multi-Agenten-Umgebungen zum Tragen.

3.1 MULTI-AGENTEN-UMGEBUNGEN

Kommunikationsfähige Agenten können sich als Agentengesellschaft bei der Lösung eines Problems gegenseitig unterstützen. Es gibt verschiedene Ansätze, wie eine Gruppe von Agenten ein (möglicherweise gemeinsames) Problem lösen kann (zu finden z. B. in [Durfee91], [Lesser90], [Albayrak93], [Sandholm95]).

Einerseits kann das Problem von einem darauf spezialisierten Agenten in Teilprobleme zerlegt werden, die danach von einem wiederum darauf spezialisierten Agenten auf die für die Teilprobleme am besten geeigneten Agenten verteilt werden. Nachdem alle Teilprobleme gelöst wurden, werden die Teillösungen wieder zu einer Gesamtlösung zusammengesetzt ([Brenner98]).

Manche Probleme sind wiederum von Natur aus verteilt und aus offensichtlichen Teilproblemen zusammengesetzt. Auch in unserem Fall (das „Problem“ DSM) setzt sich das Problem aus einer Vielzahl an gekoppelten Teilproblemen zusammen. Die einzelnen Verbraucher ergeben in Summe das eigentliche Problem bzw. das eigentlich lösbare Problem. Es existiert also eine Menge an Teilproblemen und möglicherweise eine Menge an Teillösungen. Es stellt also den natürlichen Weg dar, ein Multi-Agenten-System (MAS) für die Lösung des Problems zu verwenden.

Bei MAS kommen die „höher“ entwickelten Eigenschaften eines intelligenten Software-Agenten zum Tragen: Kommunikation und soziales Verhalten. Jeder Agent hat sein lokales, persönliches Ziel, das nicht notwendigerweise mit dem globalen, gemeinsamen Ziel übereinstimmt. Bei solchen Konflikten muß der Agent entscheiden welcher Strategie er folgt, ob er egoistisch oder sozial handeln soll. Es besteht auch die Möglichkeit, daß man das System in MAS einteilt, bei dem die lokalen Ziele implizit dem globalen Ziel in die Hände arbeiten. Hier kommt eine Eigenschaft zum Tragen, die als „fraktal“ ([Peitgen92]) bezeichnet wird: Aus den Details wird ein Ganzes. Indem jeder Agent sein persönliches Ziel anstrebt, trägt er unbewußt zum globalen Ziel etwas bei.

3.2 TYPEN VON AGENTEN

Ein Agent ist also ein autonom agierendes Programm. Das Verhalten des Agenten basiert auf seiner Architektur. Generell kann man Agenten als black box ansehen.

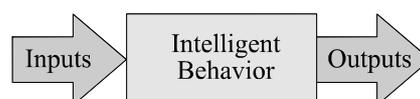


Abbildung 2: ein Agent als black box

Die Inputs bestehen aus allen sensorischen Meßwerten, die dem Agenten zur Verfügung stehen, und aus den, von anderen Agenten oder Menschen empfangenen, Nachrichten und Befehlen. Die Outputs fassen alle aktorischen Einwirkungen auf die Umwelt und alle Kommunikationsausgänge zusammen.

Ihr Verhalten teilt die Agenten in zwei Gruppen. Die erste Art wird „reaktive Agenten“ genannt ([Davis97]). Sie stellen die „einfacheren“ Agenten dar. Ein reaktiver Agent besteht aus Reflexen und Reaktionen. Er reagiert auf Veränderungen der Umwelt wozu auch die Kommunikationsumgebung zählt. Der Agent besteht aus diesem Grund aus sogenannten „Kompetenzmodulen“, die spezialisiert auf gewisse Situationen rasch auf Änderungen reagieren ([Brooks86]). Reaktive Agenten sind somit einfach und schnell.

Der zweite Typus von Agent wird als „deliberativ“ bezeichnet. Deliberative Agenten sind „intelligenter“ als reaktive. Sie verfolgen einen Plan, entwickeln Strategien um ihre Ziele zu verwirklichen und sind daher in ihrem Aufbau wesentlich komplexer. Eine interne Wissensbasis unterstützt einen sogenannten „reasoner“ bei seinen Entscheidungen über das Verhalten des Agenten ([Rao95]).

Generell kann man sagen, daß bei reaktiven Agenten die Intelligenz erst durch ein Kollektiv von Agenten und massiver Kommunikation entsteht (vgl. [Brooks91]), während der deliberative Agent an sich bereits als intelligent angesehen werden kann.

3.3 AGENTEN UND FELDBUSSE

Bevor der Autor erklärt, wie man intelligente Software-Agenten für DSM verwenden kann, soll noch die Verbindung von Agenten und der Feldbustechnologie beleuchtet werden.

Feldbusse sind im allgemeinen serielle Netzwerke, die aus einer Vielzahl von „Knoten“ bestehen. Die Knoten sind die Teilnehmer an dem Netzwerk, das z. B. die Klimaanlage eines großen Bürogebäudes steuert. Am Markt befindet sich eine Vielzahl spezialisierter wie auch universeller Feldbussysteme, die für Gebäudeautomatisierung, Fertigungsstraßen, Fahrzeugtechnik, Luft- und Raumfahrt und im Heimbereich Anwendung finden. Diese Netze können flach, hierarchisch oder gemischt aufgebaut sein ([Dietrich98]). Topologisch gibt es hauptsächlich Bus-, Ring- und Mischformen (Abbildung 3).

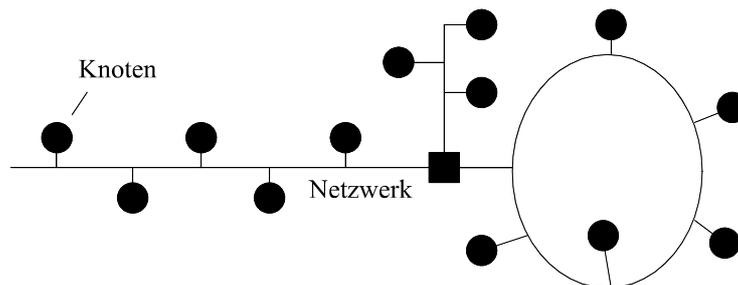


Abbildung 3: vernetzte Feldbusknoten

Die üblichen Medien sind Zweidrahtleitungen, Funk, Infrarot und Powerline. Der gemeinsame Faktor dieser unterschiedlichen Produkte ist die schmale Bandbreite des Netzwerkes und die geringe Rechenleistung der einzelnen Knoten. Für das Beispiel „Klimaanlage“ ist das auch nicht notwendig. Es reichen einfache, billige Microcontroller als Knoten und Datenraten von wenigen Kilobit um kommunizierende Thermostate zu realisieren. Die Verbindung von verteilter künstlicher Intelligenz und Feldbussen ist aber stark von bestimmten Rahmenbedingungen geprägt.

Wenn man auf den Knoten des Feldbusses Agenten beheimaten will, kann man nicht davon ausgehen die Rechenleistung für deliberative Agenten vorzufinden. Meist sind die Knoten langsame 8-bit Prozessoren mit wenigen Kilobyte an RAM und ROM ([Dietrich98]). Üblicherweise sind diese Knoten sehr schlecht für mathematische Operationen wie Gleitkommarechnungen geeignet und haben kein Datei- oder Datenbanksystem. Lediglich einfache und kurze Algorithmen können programmiert werden, meist in C oder Assembler. Die zu wählende Agentenklasse ist somit der reaktive Agent. Die Anforderungen, die ein deliberativer Agent stellt, können von einem üblichen Feldbusknoten nicht erfüllt werden.

Der reaktive Agent bringt aber eine weitere Anforderung mit sich. Meist verlangt ein MAS, das aus reaktiven Agenten besteht, einen erhöhten Kommunikationsaufwand. Das Netzwerk eines Feldbussystems ist üblicherweise paketorientiert und schmalbandig. Es werden lediglich Kommandos wie „Lampe 234 einschalten“ oder Meßwerte wie „Innentemperatur Raum 605 = 21,3°C“ verschickt. Die Pakete sind kurz und ereignisgekoppelt. Weiters muß man damit rechnen, daß Pakete verzögert werden und multiple Empfänger nicht gleichzeitig erreichen. Die Kommunikation der Agenten muß sich an diese Umstände anpassen.

Feldbusse ergeben somit einige Einschränkungen für Agentensysteme. Man kann keine großen Tabellen übertragen, mobile Agenten sind praktisch unrealisierbar, es gibt keine Datenbanksysteme und die Synchronisierung der einzelnen Agenten kann nicht immer garantiert werden.

4 AGENTEN, FELDBUSSE UND DSM

Mit den beiden Schlüsseltechnologien „Agenten“ und „Feldbusse“ soll nun eine DSM-Lösung präsentiert werden, die flexibel, billig und effizient ist.

Wenn man den Gesamtenergieverbrauch einer Gruppe an Geräten oder allgemeiner Verbraucher als globales Problem eines MAS ansieht, dann ist eines der Hauptprobleme der Agententechnik bereits gelöst: Die Dekomposition des Problems. Das DSM-Problem ist automatisch verteilt und zerlegt, wie in Abbildung 4 angedeutet ist.

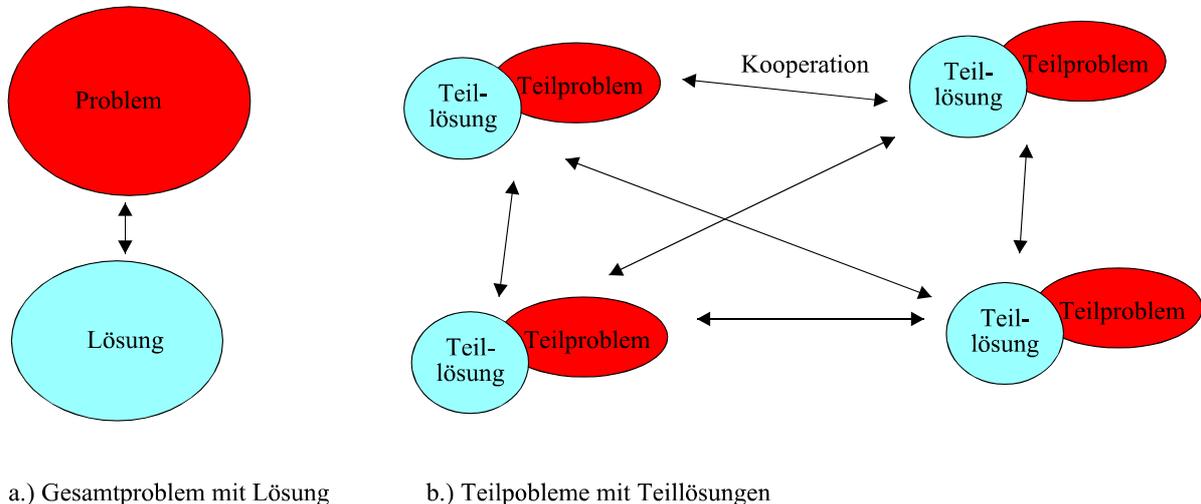


Abbildung 4: Problem-Dekomposition

Es besteht aus Teilproblemen, die lokal von den einzelnen Verbrauchern verursacht werden. Das MAS braucht sich also nicht bemühen, das globale Problem in Teilprobleme zu zerlegen. Jeder Agent kann sich um ein individuelles Teilproblem kümmern. Wenn man jedem Verbraucher einen intelligenten Agenten zuweist und Kommunikation zwischen den Agenten ermöglicht, dann erhält man ein DSM-MAS. Je nach Komplexität der Agenten können diese kooperieren, sich gegenseitig helfen oder sich lediglich mitteilen, was jedes Gerät in Zukunft beabsichtigt zu tun.

Die Teilprobleme sind also die individuellen aktuellen und zukünftigen Stromverbräuche der einzelnen Geräte. Die Teillösungen dazu sind ebenso individuell. In gewissen Betriebszuständen kann ein Gerät vielleicht kurzfristig abgeschaltet werden, etwa ein Kühlschrank. Ein anderes Gerät kann seine gesamten Aktivitäten auf später verlegen, wie eine Waschmaschine. Wenn man den einzelnen Verbrauchern bzw. deren Agenten einen gewissen Spielraum einräumt, ergeben sich viele Möglichkeiten, wie das MAS auf prognostizierte Verbrauchsspitzen reagieren kann. Wenn in einem großen Raum viele Beleuchtungskörper auf einmal eingeschaltet werden sollen, und alle eine Einschaltspitze aufweisen, so würden sich diese im Normalfall addieren. Wenn jede Lampe einen Agenten besitzt, der die Freiheit bekommt den Einschaltzeitpunkt bis zu einer Sekunde zu verzögern, so können die Agenten die Einschaltspitzen kurz nacheinander passieren lassen, so daß sie sich nicht summieren. Die Algorithmen und Regeln sind die selben wie die, die eine Waschmaschine in der Nacht schleudern lassen. Lediglich die zeitliche Granularität ist feiner. Die Agenten können auch unterschiedlich aufgebaut sein. Wenn das Kommunikations-Interface gleich ist, können aggressive Agenten mit sozialen Agenten oder Agenten mit verschiedenen Versionsnummern und Fähigkeiten in einem System arbeiten.

Wo befinden sich aber diese Agenten? Eine Möglichkeit wäre, sie auf einer Workstation mit geeignetem Agenten-Betriebssystem laufen zu lassen, die per Feldbus mit den Geräten verbunden ist.

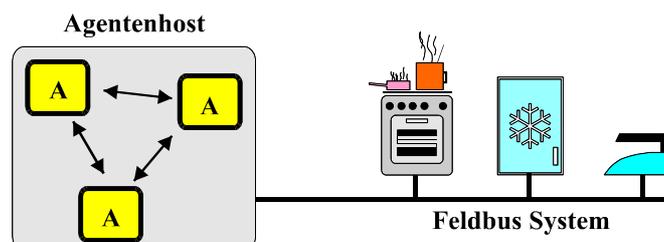


Abbildung 5: DSM-Agenten auf einem Host

Die Geräte sind hier notwendigerweise „Intelligente Geräte“, d. h. sie sind der Kommunikation mächtig. Sie kündigen, bevor sie einen Zustandswechsel eingehen, diesen Umstand und alle begleitenden Parameter wie

Stromverbrauch, Zeitdauer, etc. an. Diese Zustandswechsel können aufgrund eines geräteinternen Programms oder aufgrund von Befehlen, die über den Feldbus oder direkt auf das Gerät einwirken, erfolgen. Die Workstation wäre somit der Agentenhost (Abbildung 5). Die einzigen zwei Unterschiede zu den herkömmlichen zentralen DSM-Systemen sind der überblickbarere Aufbau des Algorithmus und die einfachere Vernetzung mit einem Feldbus. Der Aufbau ist aber immer noch zentralistisch und somit nicht fehlertolerant. Außerdem ist nach wie vor ein leistungsstarker Rechner notwendig, um den DSM-Algorithmus (der jetzt ein MAS ist) auszuführen. Die wohl vorteilhaftere Lösung ist es, die Agenten in den Feldbus zu bringen. Das Ergebnis ist eine völlig flache Hierarchie ohne einen zentralen Knoten (Abbildung 6). Da die Feldbusknoten üblicherweise einen Mikrokontroller besitzen, läßt sich auf diesem ein einfacher reaktiver, stationärer Agent ausführen.

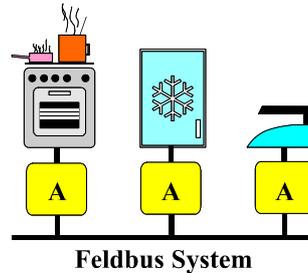


Abbildung 6: lokale DSM-Agenten durch Feldbus vernetzt

Das System ist fehlertolerant. Wenn ein Gerät ausfällt, können die anderen nicht nur weiterarbeiten, sondern auch den durch die Fehlfunktion eventuell auftretenden erhöhten Energieverbrauch kompensieren, bis der Fehler behoben ist. Ebenso fügt sich ein neues Gerät ohne Probleme in das System ein, da kein zentraler Manager benachrichtigt werden muß. Das Management ist nun verteilt ([Palensky97]). Die Kommunikation zwischen Gerät und Agent findet nun lokal statt, und belastet den Bus nicht.

Die Kooperationsmethoden eines MAS sind derzeit ein großes Forschungsgebiet und werden aufgrund ihrer Komplexität hier nicht behandelt. Es sei lediglich erwähnt, daß die Art des DSM-Problems es ermöglicht, völlig neue und effiziente Arten der Kooperation in MAS zu schaffen. Die Verteiltheit des Problems, gemeinsam mit ihren lokalen Lösungen, spielt hier eine entscheidende Rolle.

5 GLOBALE VERNETZUNG

Es sind also in intelligenten Software-Agenten und in der Feldbustechnologie Werkzeuge gefunden, um ein lokales System von Energieverbrauchern zu managen. Die Abbildungen in diesem Artikel suggerieren, daß es sich bei diesen Systemen lediglich um den Heimbereich handelt, was aber keine Einschränkung sein soll. Maschinenhallen oder Fertigungsstraßen können ebenso verwaltet werden wie kommunale Einrichtungen oder Gastronomiebetriebe. Die Frage, die sich dabei stellt ist, wo die realen Grenzen eines solchen Systems liegen. Ab welcher Knotenanzahl kann keine sinnvolle Kooperation mehr stattfinden? Das herauszufinden ist Aufgabe von Simulationen und Untersuchungen. Die Möglichkeiten und Ressourcen müssen abgewogen werden. Eine limitierte Bandbreite und Rechenleistung ergibt eine limitierte Knotenanzahl für ein MAS. Fest steht, daß man vielleicht ein Wohnhaus flach vernetzen kann, aber keine Fabrik oder Stadt. Es müssen Hierarchien eingeführt werden, wenn ein EVU an bezirks- oder bundeslandweites DSM denkt. Die globale Vernetzung eines Agentensystems bedarf einiger Infrastruktur.

Denkbar ist eine Cluster-Bildung in Form von Häuserblöcken oder kleinen Stadtvierteln. Definierte Regionen können intern DSM praktizieren, und nach außen als ein Kunde auftreten. Eine gemeinsame Schnittstelle bietet dazu die notwendige Kommunikation an. Diese Schnittstelle ist in ihrer Ausprägung ein Agent, der rein Telekommunikationsdiensten verwirklicht. Man kann über diese Schnittstelle dem Cluster Online-Tarife als DSM-Reiz anbieten oder Wünsche über bestimmte Lastkurven anbringen. Die unterlagerte Agenten-Gesellschaft kann dann versuchen diese Wünsche zu erfüllen. Die Kommunikation zwischen den einzelnen Hierarchieebenen erfolgt ebenfalls mit der Methodik von MAS. Die Cluster agieren autonom und gehen Vereinbarungen mit den Ebenen über und unter ihnen ein. Ein Cluster bzw. der Kommunikationsagent des Clusters kann z. B. das Angebot eines günstigen Tarifs in Zusammenhang mit einem speziellen Lastgang bekommen. Gemeinsam mit den DSM-Agenten des Clusters, die wiederum Kommunikationsagenten mit unterlagertem DSM-MAS sein können, wird herausgefunden, ob das Ziel erreicht werden kann. Wenn das zutrifft, kann die Vereinbarung getroffen werden.

In der klassischen Agententechnik wird diese Methode als Kontrakt-Netz bezeichnet ([Brenner98], [Albayrak93]). Ein Agent (der Kommunikationsagent) erkennt ein Problem, und schreibt es so wie es ist oder bereits zerlegt öffentlich in seinem MAS aus. Danach empfängt er die Antworten der Agenten, die sich in der Lage sehen, das Problem zu lösen, und vergibt die Aufträge. Wenn einer der empfangenden Agenten andere Agenten für die Lösung benutzt, interessiert das den ursprünglichen Auftraggeber nicht. So entsteht automatisch eine hierarchische Struktur, die stark vermascht sein kann. Ein ursprünglicher Auftraggeber kann in der Tat

später einen Teil seines Problems selber lösen, wenn er ein Teilproblem, das er ursprünglich nicht erkannt hat, annimmt. Er ist dann Auftraggeber und Auftragnehmer in einem.

In unserem Fall ist zusätzlich mit der Vermaschung noch mit einer ausgeprägten Eigendynamik zu rechnen. Wenn ein Agent sein Problem zerlegt oder löst, tauchen unter Umständen unbekannte neue Probleme auf. Eine momentane Lösung wie ein Verschieben des Schleuderganges der Waschmaschine auf später kann eine Reihe neuer Probleme aufwerfen. Die Stabilität der Kooperationsmethode ist hier die wichtigste Eigenschaft. Bei ungünstiger Wahl kommt das System zu keiner Lösung, da es sich in einem oszillierenden Zustand befindet oder die Lösung mehr Probleme aufwirft als ursprünglich vorhanden waren.

Die Möglichkeiten der globalen Vernetzung wie Internet, wireless LAN und anderer Technologien, die von ganz anderen Wirtschaftszweigen getrieben werden, kann und wird man auch für DSM nutzen. Die Kommunikationstechnik für globales und wirtschaftliches DSM wird es in nicht allzuferner Zukunft geben. Die Management-Architektur dafür ist bereits vorhanden. Komplexe und verteilte Systeme lassen sich nur mit verteilten Methoden effizient verwalten. Die Agententechnik in Verbindung mit Feldbussen ist hier der natürliche Weg dem Problem gegenüberzutreten. Nicht nur DSM ist mit Agenten in einer völlig neuen Qualität verwirklichtbar, sondern jede Art der Ressourcenverwaltung wie z. B. verteilte Energieversorgung. Die Autonomie von MAS ist eines der Hauptargumente für den Einsatz dieser Technologie. Einmal mit einem Auftrag versehen, kümmert sich der Agent um alles weitere. Diese Eigenschaft ist notwendig, wenn man sich sehr komplexen Systemen widmet. Und DSM kann bereits im kleinen Rahmen sehr kompliziert werden.

Janpieter Scheerder von SunSoft hat den Sun-Slogan „the network is the computer“ auf der LonUsers-Konferenz 1997 in Amsterdam in „the network is the control“, umgewandelt, um auf verteilte Steuerungssysteme hinzuweisen. In Zukunft wird es wohl auch heißen „the network is the management“.

Literatur

- [Albayrak93] Sahin Albayrak and Stefan Bussmann: “Kommunikation und Verhandlungen in MAS”, in Jürgen Müller (editor): “Verteilte künstliche Intelligenz”, BI-Wissenschafts-Verlag, ISBN 3-411-16181-7, 1993
- [Brenner98] Walter Brenner, Rüdiger Zarnekow, Hartmut Wittig: “Intelligente Software Agenten”, ISBN 3-540-63431-2, Springer, 1998
- [Brooks86] Rodney A. Brooks: “a robust layered control system for a mobile robot”, in IEEE Journal of Robotics and Automation, vol. RA-2(1), April 1986
- [Brooks91] R. A. Brooks: “Intelligence without representation”, in Artificial Intelligence Vol. 47, pp139-159, 1991
- [Davis97] Darryl N. Davis: “Reactive and Motivational Agents: Towards a Collective Minder”, in Jörg P. Müller, Michael J. Wooldridge and Nicholas R. Jennings (editors): “Lecture Notes in Artificial Intelligence 1193, Intelligent Agents III, Proceeding of ECAI’96 Workshop”, ISBN 3-540-62507-0, Springer, 1997
- [Dietrich98] D. Dietrich, D. Loy und H.-J. Schweinzer (Hrsg.): „LON-Technologie“, Hüthig Buch Verlag GmbH, ISBN 3-7785-2581-6, Heidelberg, 1998
- [Durfee91] E. H. Durfee and V. R. Lesser: “Partial global planning: A coordination framework for distributed hypothesis formation”, in IEEE Transactions on Systems, Man and Cybernetics 21, 1991
- [Franklin97] S. Fanclin and A. Graesser: “Is it an agent, or just a program? A taxonomy for autonomous agents”, in Jörg P. Müller, Michael J. Wooldridge and Nicholas R. Jennings (editors): “Lecture Notes in Artificial Intelligence 1193, Intelligent Agents III, Proceeding of ECAI’96 Workshop”, ISBN 3-540-62507-0, Springer, 1997
- [Hayes95] B. Hayes-Roth: “An Architecture for Adaptive Intelligent Systems”, in Artificial Intelligence: Special Issue on Agents and Interactivity, (72), pp329-265, 1995
- [Lesser90] V. R. Lesser and K. Decker: “Extending the partial global planning framework for cooperative distributed problem solving network control”, Technical Report COINS Report, University of Massachusetts, Amherst, 1990
- [Palensky97] P. Palensky, D. Dietrich, R. Posta und H. Reiter: „Demand Side Management by Using LonWorks“, in Proceedings of the 1997 IEEE International Workshop on Factory Communication Systems, ISBN 0-7803-4182-1. Barcelona, 1997
- [Peitgen92] H. O. Peitgen, H. Jürgens and D. Saupe: “Fractals for the Classroom”, Springer Verlag, 1992
- [Petrie97] Charles Petrie: “What is an Agent?”, in Jörg P. Müller, Michael J. Wooldridge and Nicholas R. Jennings (editors): “Lecture Notes in Artificial Intelligence 1193, Intelligent Agents III, Proceeding of ECAI’96 Workshop”, ISBN 3-540-62507-0, Springer, 1997
- [Rao95] A. S. Rao, M. P. Georgeff: “BDI Agents: From Theory to Practice”, in Proceeding of

the 1st International Conference on Multi-Agent-Systems (ICMAS), San Francisco, 1995

[Sandholm95]

T. Sandholm and V. Lesser: "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework", Proceedings of the First International Conference on Multi-Agent-Systems (ICMAS), San Francisco 1995