


```
transition_keyPresentTrunk_8752();
transition_noCommand_8757();
transition_nonOffCommand_8758();
transition_true_8761();
transition_true_8762();
// DangerDoorStatus
transition_noCommand_8763();
transition_nonOffCommand_8764();
transition_true_8767();
// DangerDoorStatus
transition_true_8768();
transition_activate_8735() {
  test
  (F_VALID_SensingControl_8721_carCommand == true)) {
  // for tests
  SensingOnCmd(INPUT_PORT_SensingControl_8721_carCommand)) {
  current_state_SensingControl_8721 = STATE_KeyAbsent_8728;
  UT_PORT_SensingControl_8721_cabinSensingOn = SA_On;
  UT_PORT_VALID_SensingControl_8721_cabinSensingOn = true;
  UT_PORT_SensingControl_8721_trunkSensingOn = SA_On;
  UT_PORT_VALID_SensingControl_8721_trunkSensingOn = true;
  UT_PORT_SensingControl_8721_sensingMode = SM_KeyAbsent;
  UT_PORT_VALID_SensingControl_8721_sensingMode = true;
  true;
```

Link
www4.in.tum.de

Intelligente Maschinen ohne Denkfehler

Intelligente Steuerungseinheiten finden sich heute in allen Geräten – vom Mobiltelefon über das Auto bis zu Produktionsanlagen. Die einzelnen Systeme müssen nicht nur für sich alleine zuverlässig funktionieren, sondern auch ohne Missverständnisse zusammenarbeiten. Dafür entwickeln TUM-Softwarespezialisten mathematische Modelle

Schon auf der Fahrt zur TUM begeben sich die Hände intelligenter Systeme. Unter der Motorhaube, hinter dem Armaturenbrett und an etlichen anderen Stellen werden unermüdlich Informationen über die Umgebung, meinen Fahrstil und das Verhalten des Autos gesammelt, ausgewertet und es wird entschieden, was zu tun ist. Das kann die Klimaanlage sein, die für angenehme Temperaturen sorgt, das können aber auch mehrere Aufpasser sein, die gemeinsam damit beschäftigt sind, das Auto bei gewagten Manövern sicher in der Spur zu halten. Eingebettete Software, „Embedded Systems“, verleiht fast allen modernen Maschinen die Fähigkeit zum Mitdenken.

Klein, autark und schwer zu beherrschen

Embedded Systems bestehen aus Sensoren, die Informationen liefern, Aktuatoren, die diese in Aktionen umsetzen, Speicherbausteinen, programmierbaren Prozessoren und vor allen Dingen aus Software. Einmal losgelassen, machen sie stur, was ihnen die Befehle der Entwickler vorgeben. Weil sie so autark sind, müssen Embedded Systems ganz besonders zuverlässig arbeiten. Das abzusichern, ist nicht einfach. Allein in einem Mittelklassewagen sind heute bis zu 50 Steuergeräte am Werk, die zum Teil von unterschiedlichen Herstellern stammen und mit unterschiedlicher Software arbeiten. Da kann es zu Überraschungen kommen, obwohl jede Einheit für sich perfekt funktioniert. „Die Elektronik spinnt“, sagt der Laie dann einfach. Von „Feature In-

teraction“ spricht der Systemfachmann. Tatsache ist: Zwischen zwei oder mehr Steuergeräten ist ein Missverständnis aufgetreten.

Am Lehrstuhl für Software & Systems Engineering der TU München sitzen Martin Feilkas und Dr. Bernhard Schätz am Tisch und konstruieren so ein Missverständnis, wie es – wohlgemerkt theoretisch – entstehen könnte. Man stelle sich vor, ein Wagen mit Automatikgetriebe wird mit offener Tür und laufendem Motor abgestellt. Das Getriebe steht auf Parkposition und trotzdem fährt das Auto plötzlich los. Wie das geht? „Eine klassische Feature Interaction“, sagt Bernhard Schätz und lässt seine Gedanken spielen. „Es ist Sommer und weil die Tür offen ist, wird es warm im Auto. Die Klimaanlage fängt an zu kühlen und zieht Strom aus der Batterie. Der Motor dreht hoch, um die Batterie zu versorgen. Nun ist in älteren Systemen der Vorgang „Motor hochdrehen“ so gelöst, dass sich das Gaspedal automatisch bewegt. Das aber bedeutet, die Feststellbremse zu lösen und auf Fahrposition umzustellen. Der Wagen fährt – scheinbar selbstständig – los.“ Ein Mechaniker, der den Fehler klären wollte, würde weder schadhafte Teile finden noch könnte er zurückverfolgen, welches Ereignis oder welcher Zustand genau diesen Ausfall verursacht hat. Um trotzdem eine hohe Zuverlässigkeit zu erreichen, stecken die Hersteller komplexer Systeme etwa 40 Prozent der Entwicklungskosten in Testläufe, die alle eventuell eintretenden Ereignisse und die möglichen Reaktionen aller beteiligten Komponenten abprüfen. ▶

SPES 2020

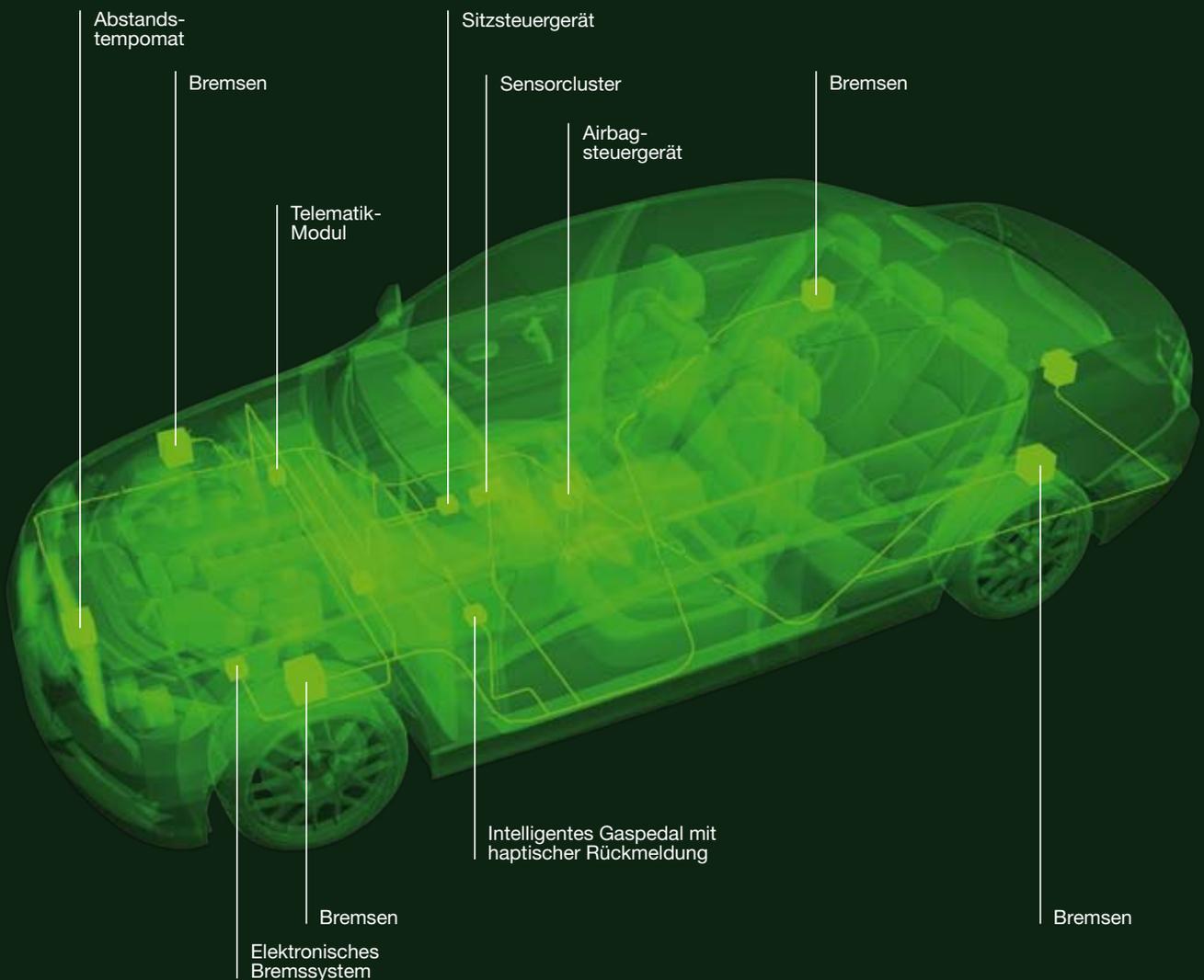
Kaum ein anderer Markt wächst so stabil wie der für Embedded Systems. Eine Studie des Bundesverbandes Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM) aus dem Jahr 2008 sagt für den deutschen Markt in den kommenden Jahren ein durchschnittliches Wachstum von 9 bis 10 Prozent voraus. Das weltweite Marktvolumen wird auf 60 Milliarden Euro geschätzt. Das Bundesministerium für Bildung und Forschung (BMBF) fördert die Entwicklung von „Software für Embedded Systems“ im Rahmen der Innovationsallianz SPES 2020 mit insgesamt 23 Millionen Euro über eine Laufzeit von drei Jahren. SPES 2020 ist ein Verbund von derzeit 21 Partnern aus Wirtschaft und Wissenschaft, vertreten jeweils durch einen Sprecher. Professor

Manfred Broy vom Lehrstuhl für Software & Systems Engineering der TU München spricht für die akademischen Mitglieder der Allianz, Dr.-Ing. Reinhold Achatz, Leiter der Forschung bei Siemens Corporate Technologies, vertritt die Industriepartner. SPES 2020 besteht aus einem Zentralprojekt, in dem Methoden und mathematische Grundlagen erarbeitet werden, und den fünf Anwendungsgebieten Avionik, Medizin, Energie, Automobil und Automatisierung. In diesen Branchen ist Deutschland für die hohe Qualität seiner Produkte bekannt. Weil Software dort mittlerweile der wichtigste Innovationsfaktor ist, sollen die hohen Ansprüche an Produkte „Made in Germany“ in Zukunft auch an Embedded Systems gestellt werden können.

```

TYPE_boolean fire_transition_modePresent_8746() {
    __current_state_SensingControl_8721 = STATE_KeyPresent_8731;
    OUTPUT_PORT_SensingControl_8721_sensingMode = SM_KeyPresent;
    OUTPUT_PORT_VALID_SensingControl_8721_sensingMode = true;
    return true;
}
TYPE_boolean fire_transition_noCommand_8747() {
    // empty port test
    if ((INPUT_PORT_VALID_SensingControl_8721_carCommand == false)) {
        if (fire_transition_modeInactive_8744()) {
            return true;
        }
    }
    return false;
}
TYPE_boolean fire_transition_true_8748() {
    if (fire_transition_deactivate_8736()) {
        return true;
    }
}

```



Bis zu 50 Steuergeräte sind heute schon in einem Mittelklassewagen am Werk – zum Beispiel für Fensterheber, Klimaanlage oder Motorsteuerung. Viele Fahrerassistenzsysteme greifen auf Informationen von mehreren Steuergeräten zurück. Die modellbasierte Entwicklung soll sicherstellen, dass alle Steuergeräte in allen möglicherweise auftretenden Situationen fehlerfrei miteinander kommunizieren

Das Bedürfnis nach einer effizienteren Methode ist groß. Die vom Bundesministerium für Bildung und Forschung mit 23 Millionen Euro geförderte Innovationsallianz SPES 2020 will das Thema angehen. In SPES vertreten sind Hochschulen, Forschungsinstitute sowie Unternehmen aus fünf Schlüsselbranchen. Ihr Ziel ist ein Designprozess, der die Sicherheit von Embedded Systems bereits im Voraus garantiert. Professor Manfred Broy vom Institut für Informatik an der TU München ist einer der beiden Sprecher des Verbunds.

Wie sieht ein Softwareentwickler alle Eventualitäten voraus, die er absichern muss, damit sein Produkt zuverlässig ist? Embedded Systems sind heute so komplex, dass unsere Vorstellungskraft dafür nicht mehr ausreicht. Bernhard Schätz rechnet vor: „Bei den real vielen Tausend Zuständen pro Steuergerät ist das mit Testen alleine nicht zu bewältigen.“ Broys Team und die Kollegen aus dem SPES-Zentralprojekt sehen die Lösung in Modellen. Sie sollen die wesentlichen Eigenschaften eines Systems beschreiben und die Grundlage für seinen schrittweisen Aufbau bilden.

Bernhard Schätz und Martin Feilkas vergleichen das Modell mit dem Bauplan für ein Haus. Wie der Installateur, der Elektriker und der Maurer auf verschiedene Sichten des gleichen Plans zurückgreifen, sollen in Zukunft die Informationen aus verschiedenen Entwicklungsbereichen – Software, Elektrotechnik und Mechanik – in ein gemeinsames Grundmodell einfließen. Heute arbeiten die drei Disziplinen relativ getrennt voneinander. Das führt dazu, dass man oft erst im fertigen Produkt testen kann, ob alle Systeme zuverlässig zusammenspielen. Mit dem modellbasierten Ansatz kann man schon während der Konstruktion einzelne Funktionen überprüfen und möglichen Fehlern gegensteuern.

Genau wie ein Bauplan auf Vorschriften basiert, die garantieren, dass das Haus sicher konstruiert ist, brauchen auch die Modelle Regeln, die ihre Korrektheit gewährleisten. Die Wissenschaftler wollen ein Verfahren schaffen, das diese Sicherheit mit eingebaut hat. Dafür greifen sie auf die Mathematik zurück, denn sie bietet das elegante Mittel des Beweises, der die Korrektheit eines – beliebig komplexen – Systems einwandfrei belegt. Die Forscher erarbeiten zunächst mathematische Grundlagen, aus denen dann Regeln für die Konstruktion von Modellen entstehen.

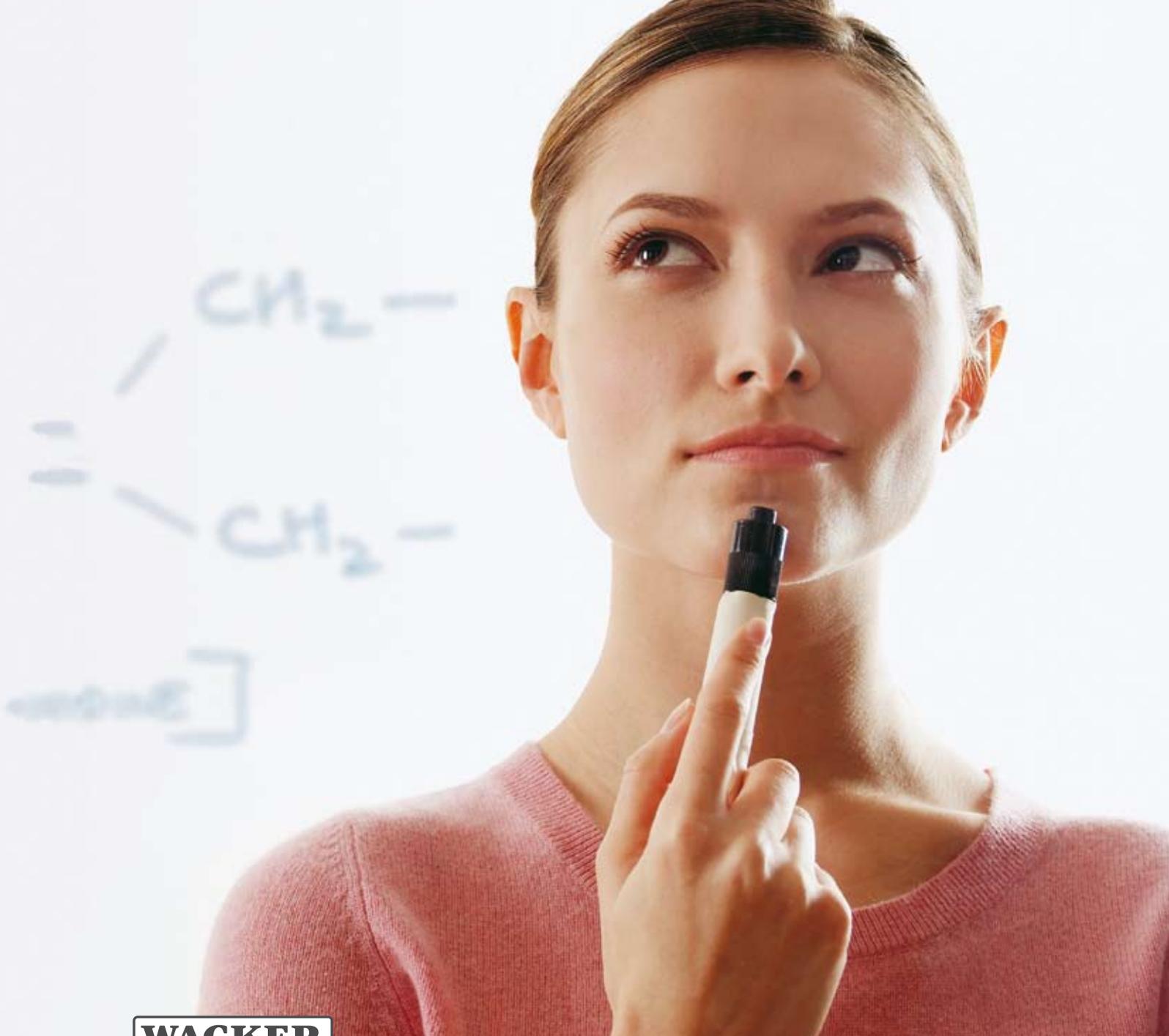
Eine große Herausforderung ist dabei die Entwicklung von Modellsprachen. Wie stellt man ein komplexes System so dar, dass mögliche Fehler nicht übersehen werden können? Die Frage kommt nicht von ungefähr, denn die Software eines Steuergeräts ist in einer Programmiersprache geschrieben. Sie besteht aus vielen Zeilen von Anweisungen (Code), aber genau wegen dieser Vielzahl an Einzelbefehlen sieht man nicht mehr, was das System eigentlich tut. Das macht es so

schwierig, mögliche Fehler zu erkennen. Besser ist es, die Systeme in einer Art und Weise aufzuschreiben, die ihre Funktion adäquat und präzise darstellt.

Ein abgesicherter Baukasten

Die Wissenschaftler wollen einen Baukasten aufeinander aufbauender Werkzeuge schaffen. Als Fundament sichern mathematische Grundlagen die Korrektheit aller nachfolgenden Bausteine ab. Sie bilden das Regelwerk, nach dem Modelle und passende Beschreibungstechniken entwickelt werden. Standardisierte Werkzeuge, die auf dieselben Regeln zurückgehen, erstellen daraus den Programmcode für die einzelnen Systeme. Am Ende stehen Analyse- und Prüfmethode zum Testen der Software. Damit wird der Softwareentwickler belegt, dass sein Produkt zu hundert Prozent zuverlässig ist, indem er auf das zugrunde liegende Modell und dessen bewiesene Korrektheit verweist. ▶

```
    }
    }
    return false;
}
#include „SensingControl_8721.h“
TYPE_boolean fire transition deactivate_8736
// non-empty // ports
TYPE_KeyType INPUT_PORT_S
if ((INPUT_PORT_VALID == INPUT_PORT_VALID)
// disconnected
TYPE_KeyType INPUT_PORT_S
if (is_CabinPresent)
TYPE_boolean INPUT_PORT_VALID
TYPE_SenseActivationType OUTPUT_PORT_VALID
TYPE_boolean OUTPUT_PORT_VALID
TYPE_SenseActivationType OUTPUT_PORT_VALID
TYPE_boolean OUTPUT_PORT_VALID
TYPE_CarCommandType INPUT_PORT_VALID
TYPE_boolean INPUT_PORT_VALID
}
TYPE_int INPUT_PORT_SensingControl
TYPE_boolean INPUT_PORT_VALID
return false;
TYPE_SensingModeType OUTPUT_PORT_VALID
TYPE_boolean OUTPUT_PORT_VALID
TYPE_boolean fire transition keyPresentCabin
// non-empty // resets output buffers
void clear_output_SensingControl
OUTPUT_PORT_VALID Ser
&& (OUTPUT_PORT_VALID Ser
// precondition
OUTPUT_PORT_VALID Ser
if ((kt_key == INPUT_PORT_SensingControl
== INPUT_PORT_SensingControl
if (fire transition noCommand_87
// symbolic states
#define STATE_Inactive_8725 0
```

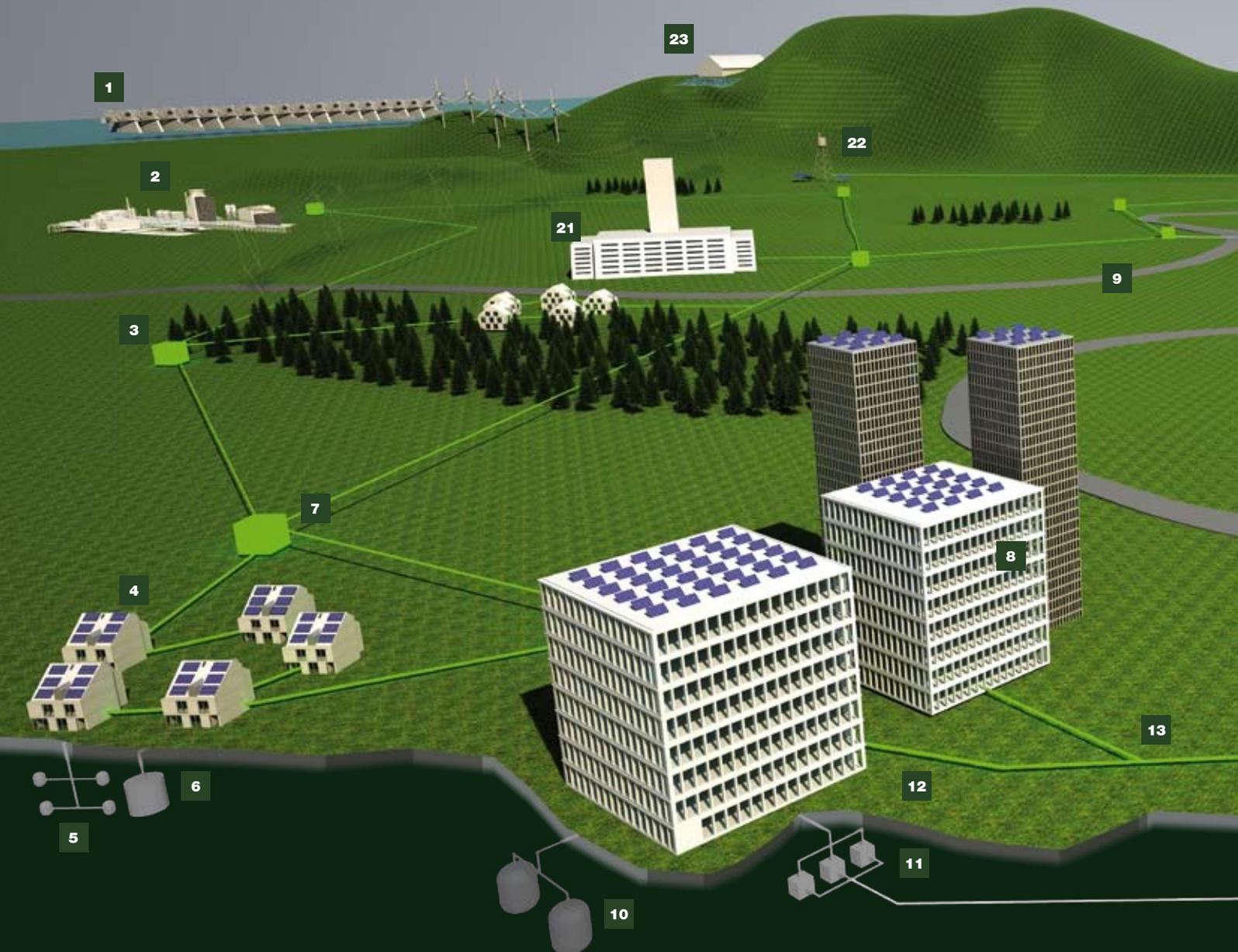


WACKER

FÜR FORSCHER, DIE DIE PRODUKTE VON MORGEN ENTWICKELN

Die Experten von morgen sind die Studenten und Doktoranden von heute. Deshalb unterstützen wir die Forschung an Universitäten – durch die Stiftung des WACKER-Lehrstuhls für Makromolekulare Chemie an der TU München, durch Kooperationen mit über 60 Universitäten und durch Praktika und Werkführungen.

CREATING TOMORROW'S SOLUTIONS



1 Wasserkraftwerk 2 Kraftwerk 3 Transformator 4 Solarzellen 5 Mikrostromspeicher 6 Thermische Speicher
 7 Energieübertragung 8 Private Kraft-Wärme-Kopplung 9 Solarzellen 10 SMES (Supraleitende Magnetische Energiespeicher)
 11 Brennstoffzellen 12 Kraft-Wärme-Kopplung

Modelle für intelligente Stromnetze

Die TUM arbeitet im Rahmen von SPES mit mehreren Industriepartnern zusammen. Bei einem der Projekte geht es um die Entwicklung intelligenter Stromnetze, ein Vorhaben, an dem die Forschungsabteilung von Siemens (Corporate Technology, CT) und die Stadtwerke München beteiligt sind. Die Siemens-Forscher Klaus Beetz und Dr. Richard Kuntschke erklären, worum es geht: „Europas Stromnetze sollen fit werden für einen Energiemarkt, zu dem neben Großkraftwerken auch viele dezentrale Produzenten von erneuerbaren Energien beitragen.“ Der Anteil der Kleinerzeuger ist aber schlecht planbar, weil er unter anderem vom Wetter ab-

hängt. Hinzu kommt, dass es in Zukunft intelligente Geräte geben wird, die sich zum Beispiel innerhalb eines vorgegebenen Zeitfensters eigenständig genau dann einschalten, wenn viel – und damit billiger – Strom im Netz vorhanden ist.

Die Stadtwerke stehen als Energieversorger vor der Aufgabe, diese zunehmende Komplexität und Dynamik zu berücksichtigen und gleichzeitig die Stabilität der Netze und die Versorgungssicherheit zu gewährleisten. In Zukunft soll ein Smart Grid intelligente Schnittstellen zwischen Kraftwerken, Energieversorgern und Kleinerzeugern schaffen. Diese sogenannten IKT-Gateways machen die Energie erzeugenden Verbraucher zu akti-



19

20

16

18

15

14

13 Verbrauchsmanagement **14** Wasserstoffspeicher **15** Wasserstoffproduktion und Tankstation
16 Wellenenergie **17** Wettervorhersage **18** Mikronetz **19** Windkraft-Anlagen vor der Küste und auf hoher See
20 Biogasanlage **21** Örtliches Kontroll- und Kommunikationszentrum **22** Solarpark **23** kleines Wasserkraftwerk

ven Teilnehmern im Netz, die sowohl selbst steuern als auch gesteuert werden können. Die Gateways sorgen für die Einbindung der zukünftigen intelligenten Erzeugungs-, Verbrauchs- und Zählereinheiten in den einzelnen Haushalten in das Smart Grid.

Modellbasierte Entwicklung bietet sich für diese Aufgabe gleich aus mehreren Gründen an. Zum einen gibt es noch keine existierenden Systeme, auf denen man aufsetzen könnte. Zum anderen haben die Entwickler festgestellt, dass sie mit ihren herkömmlichen Programmiersprachen nicht alle Eigenschaften eines solch weiträumig verteilten, autonomen Systems beschreiben können. Richard Kuntschke sieht die Knackpunkte

unter anderem in der adäquaten Darstellung von Dynamik und Wahrscheinlichkeiten. Damit meint er den Umstand, dass im gesamten Smart Grid Hunderttausende bis zu Millionen Steuergeräte existieren, die nicht immer aktiv präsent sind, die jederzeit neu hinzukommen oder wegfallen können und die mit einer gewissen Wahrscheinlichkeit ausfallen. Dann muss sichergestellt sein, dass alle Informationen trotzdem noch ankommen. Das Ziel ist deshalb, eine Modellierungstechnik zu finden, die solche Szenarien abbilden kann. Kuntschke geht davon aus, dass nach der Projektlaufzeit von drei Jahren die ersten Prototypen getestet werden können.

Christine Rüth