

Zeitschrift: Comtec : Informations- und Telekommunikationstechnologie =
information and telecommunication technology

Band: 76 (1998)

Heft: 4

Artikel: TIMS : ein TMN-Labor in der Westentasche

Autor: Eberhardt, Rolf

DOI: <https://doi.org/10.5169/seals-877291>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. [Siehe Rechtliche Hinweise.](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. [Voir Informations légales.](#)

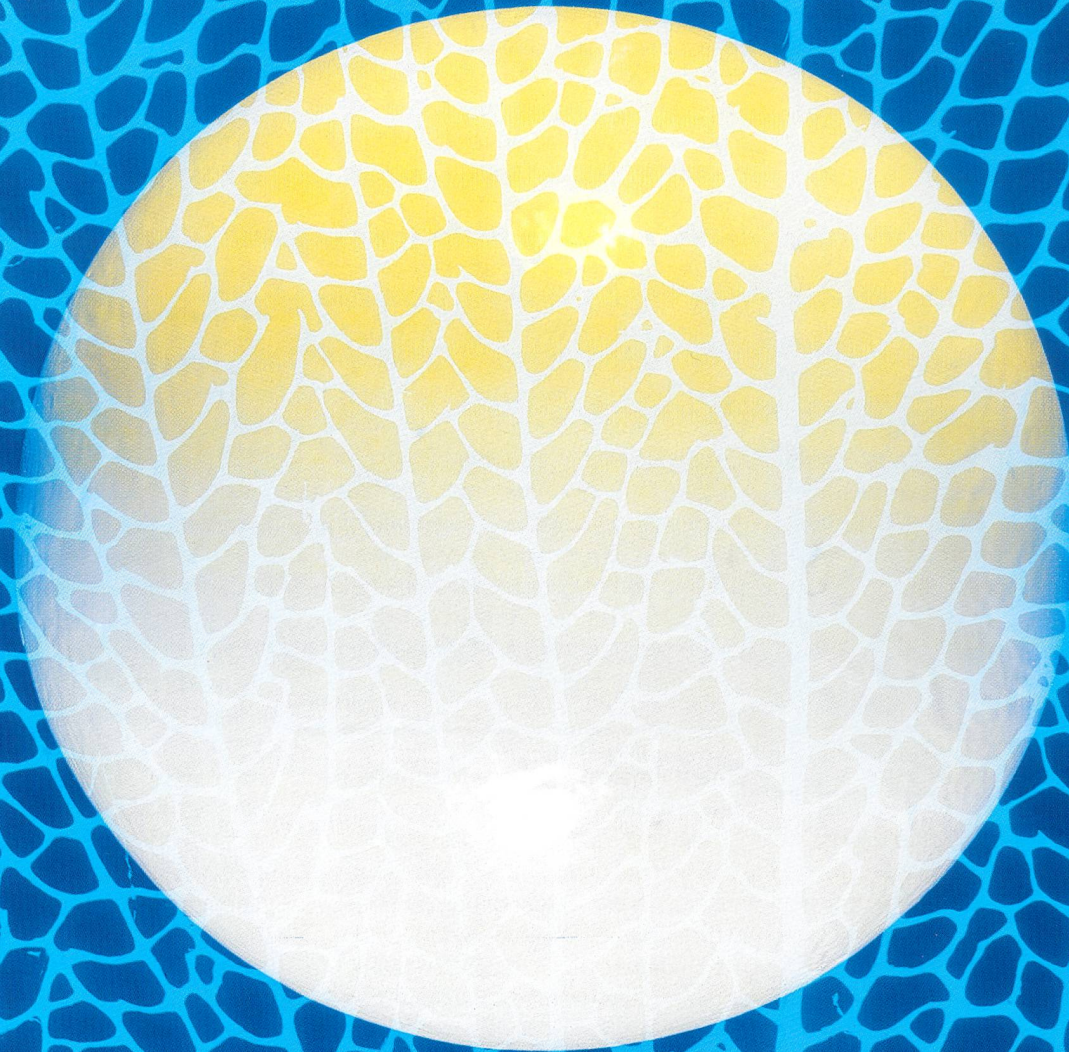
Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. [See Legal notice.](#)

Download PDF: 17.11.2024

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Aus den Explorations-Programmen von Corporate Technology (6)



TIMS – ein TMN-Labor in der Westentasche

Das Explorations-Programm* EP97-6 «Operational Processes & Customer Care» untersucht den Einsatz neuer Technologien auf die Prozesse «Service Order & Delivery» und «Service Assurance» unter den Aspekten der Dienstentwicklung und der Kosteneffizienz.

* Explorationsprogramme werden von Corporate Technology im Auftrag der Konzernleitung durchgeführt und regelmässigen Reviews unterzogen. Die Aktivitäten haben einen mittel- bis langfristigen Zeithorizont, 2–7 Jahre, je nach Gebiet.

Lassen sich durch Mehraufwand bei der Spezifikation die Einführungskosten eines TMN-Systems gesamthaft senken und der aufwendige Prozess von Spezifikation bis Test und Inbetriebnahme verkürzen? Vermutlich ja. Ein Teil der Antwort findet sich im TMN-Informationsmodell-Simulator TIMS, entwickelt von Corporate Technology und dem Institut Eurécom. In TIMS kann der Designer innert kurzer Zeit lauffähige TMN-Agenten und -Manager entwerfen und auf interne Konflikte überprüfen. Obwohl ursprünglich für den Einsatz im TMN-Bereich vorgesehen, erlaubt TIMS den Bau beliebiger Objektsimulationen, weswegen er nebst Q3 auch Komponenten für die CORBA-Schnittstellenspezifizierung enthält. TIMS bietet aber auch Investitionsschutz, indem die Resultate der Prototypentwicklung als Grundlage für die Beschaffung sowie für die Erzeugung von Testsuiten verwendet werden können.

Vor vier Jahren entschied Swisscom SDH-Netztechnologie einzuführen, geliefert von zwei Ausrüstungsherstellern. Nebst besserer Übertragungstechnologie sollte integriertes Netzmanagement entscheidende Vorteile

ROLF EBERHARDT, BERN

bei der Bewirtschaftung dieses Netzes bringen. Ähnliche Überlegungen machte man auch für das ISDN-Netz. In beiden Fällen sollte das TMN – Telecommunications Management Network – Konzept [1] zum Einsatz kommen. Wesentlich für den Erfolg von TMN sind die Schnittstellen zwischen den Systemkomponenten der verschiedenen Hersteller. Nur wenn die Funktionsweise dieser Schnittstellen klar definiert ist, lassen sich die Komponenten ohne allzu grosse Schwierigkeiten miteinander verbinden. Sind sie aber unzureichend definiert, muss der Hersteller die Anforderungen interpretieren und läuft Gefahr, eine inkompatible Komponente zu entwickeln. Oft erkennt der Netzbetreiber diese Fehler erst im Integrationstest, was zu empfindlichen Verzögerungen und Kostenüberschreitungen führen kann. Kein Wunder also, dass TMN-basiertes Netzmanagement als eigentlicher Albtraum der Netzbetreiber gilt.

Rapid-Prototyping und TMN

Eine Laborumgebung, in der ohne grosse Kosten die Anwendung simuliert sowie die Integrationsprobleme untersucht werden, kann Abhilfe schaffen. Die Internet Engineering Task Force sowie verschiedene Industrieforen (z. B. ATM Forum oder OMG) arbeiten nach dem Prinzip, dass ein Standard nur nach eingehenden Laborversuchen publiziert werden kann. Wohl existieren TMN-Toolkits, doch sind sie nicht geeignet, um rasch und ohne grosse Kosten TMN-Systeme zu entwerfen.

Ausgehend von dieser Erkenntnis entwickelte Corporate Technology zusammen mit dem Institut Eurécom ein virtuelles TMN-Labor, den TMN-Informationsmodell-Simulator TIMS. TIMS ist eine Prototyping-Umgebung für TMN-Agenten und TMN-Manager. GDMO und GRM beschreiben die Objektstruktur sowie die Objektrelationen, während das Verhalten in der LISP-basierten Sprache *Scheme* programmiert wird. Via spezielle Bibliotheken kann der Entwickler CMIP oder CORBA IIOP-basierte Interaktionen mit anderen TMN-Komponenten bauen. Die Entwicklungsumgebung sowie die Simulatorsteuerung sind vollständig in den Programmiereditor *emacs* eingebettet. Dies, sowie der eingebaute *Scheme*-Interpreter ermöglichen kurze Entwicklungszyklen, ähnlich einem Microsoft Visual Basic.

Nebst dem Run-time debugger ermöglicht ein Visualisierungswerkzeug die grafische Darstellung des Management Information Base (MIB) – also der Datenbank des TMN-Agenten zur Laufzeit. Der Anwender sieht, wo und wie sich die Objekte und ihre Beziehungen zueinander verändern und kann durch einfaches Anklicken den Inhalt der MIB-Objekte anwählen.

Eine lauffähige Simulation eines Agenten oder Managers wird TIMS-Box genannt. Die TIMS-Box wird entweder durch Szenariendateien, die bestimmte Abläufe beschreiben, gespeist, oder sie kommuniziert via Schnittstellen mit externen Systemen (Bild 1).

Vom Prototyp via Spezifikation zur Implementierung

Rechtfertigen diese Argumente die Entwicklung einer eigenen Plattform? Die Systemführung lässt sich in die drei Phasen Spezifikation, Implementierung und Inbetriebnahme unterteilen. Weil jede Phase meistens durch andere Experten begleitet wird, geht beim Sprung von Phase zu Phase applikationspezifisches Know-how verloren. Eine Spezifikation würde ungleich an Wert gewinnen, wenn das beim Verfassen erarbeitete Wissen auf einfache Art und Weise bei den nachfolgenden Phasen wiederverwendet werden könnte (Bild 2).

TIMS soll möglichst viele dieser Hürden bewältigen helfen. So lässt sich aus dem TIMS-Objektmodell und dem Objektverhalten automatisch eine Spezifikation generieren, die als *Beschaffungsgrundlage* verwendet werden kann. Zudem entspricht sie im Inhalt der neuen ODP-basierten Methode, wie ITU-T Standards schreibt, was die Akzeptanz zusätzlich erhöht.

Software-Spezifikationen sind immer fehlerbehaftet, oft infolge von Inkonsistenzen. Wie lassen sie diese erkennen? Nebst dem normalen Code programmiert der Entwickler zusätzliche Regeln, die zu jeder Zeit vom System erfüllt werden müssen. Diese Randbedingungen (Pre- und Post-Conditions sowie Invarianten) werden während der Laufzeit stetig überprüft. Besteht eine Spezifikation den *Validationstest*, kann behauptet werden, dass in der Spezifikation keine internen Konflikte bezüglich der im Test verwendeten Szenarien bestehen. Ausgestattet mit Q3- und CORBA-Schnittstellen kann ein TIMS-Prototyp als

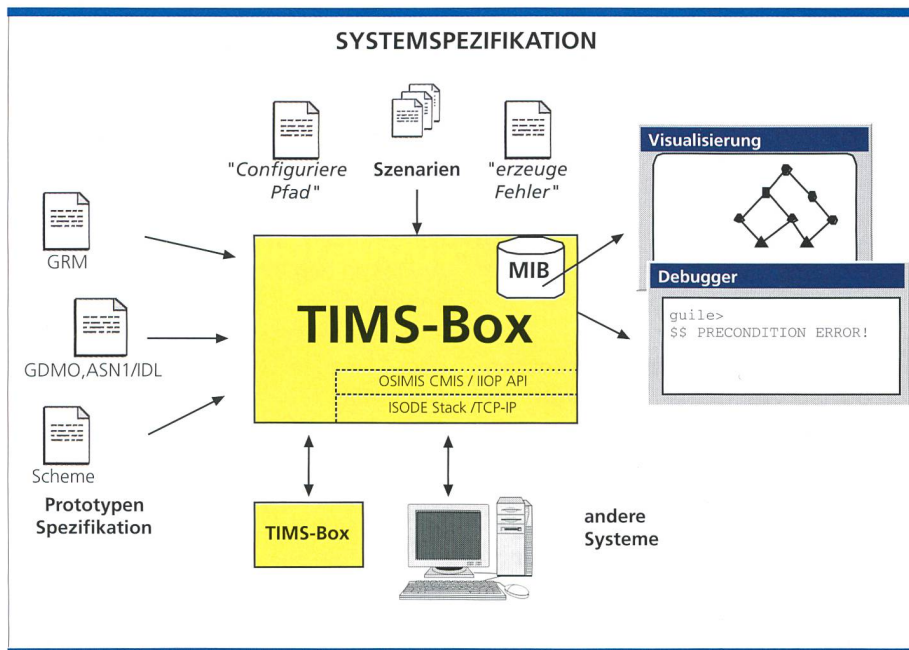


Bild 1. Nach erfolgter Spezifikation der TMN-Komponente lässt sich das Verhalten des Systems anhand von Szenarien durchspielen. Debugger und Visualisierungswerkzeuge ermöglichen Einblick in die Abläufe im Systeminneren. Via Schnittstellen kann TIMS mit anderen Systemen kommunizieren.

Referenzkonfiguration und zur Emulation in TMN-Architekturen eingesetzt werden, beispielsweise wurde das Management eines europäischen SDH-Netzes nachgebaut.

Als besondere Herausforderung gelten der Factory Acceptance Test (FAT) und der Integrationstest von Systemen. Ausgehend von der Spezifikation, entwickelt der Testspezialist Szenarien, die die Korrektheit des Produktes prüfen. Erste Experimente haben gezeigt, dass aus dem Objekt- und Verhaltensmodell sowie den Szenarien komplexe Testsuiten automatisch generiert werden können. Dies reduziert den Vorbereitungsaufwand enorm, wodurch sich der Testspezialist auf komplexe Tests konzentrieren kann.

TIMS-Boxen entwerfen

Um eine lauffähige Simulation zu erhalten, muss der Designer zwei Komponenten definieren: das Objektmodell sowie das Objektverhalten. Entity-Relationship Diagramme gelten als Grundlage für die Modellierung von datenbasierten Objektmodellen. Während Relationen in der Modellierung bestens bekannt sind, werden sie bei der Implementierung oft durch primitive Konstrukte wie Pointer ersetzt. Der Code büsst dadurch an Stabilität ein, denn

ohne Relationen lassen sich Designentscheidungen wie die Anzahl oder Art der Objekte, die involviert sind, zur Laufzeit nicht mehr überprüfen. TIMS setzt auf

Relationen, um zwischen Objekten zu navigieren. Beschrieben werden Relationen in GRM [2], das sich stark an die Objektdefinitionssprache GDMO [3] anlehnt. GRM bietet auch Operatoren zur Verwaltung von Relationen und Rollen an. Die ESTABLISH-Operation, zum Beispiel, erstellt eine Relationsinstanz, und durch eine BIND-Operation können Objekte an eine bestehende Relation angehängt werden.

Das Objektverhalten oder «behaviour» stellt eine der Knacknüsse in der Software-Entwicklung dar. Applikationschnittstellen-Beschreibungen (API) beschreiben zwar die Struktur der Schnittstelle und Information, jedoch nicht, wie diese reagiert. Meistens begnügen sich die Entwickler mit einer Beschreibung in Prosa, beispielsweise «Fällt in einer Ausrüstung (equipment) die Speisung aus, müssen deren Status (operationalState) und jene der darauf aufgesetzten Dienste (terminationPoints) von «enabled» auf «disabled» schalten». Selbst ein sehr vorsichtiger Designer kann jedoch die Möglichkeit von Nebeneffekten (sideeffects) durch andere Verhaltensregeln nicht ausschließen. So könnte beispielsweise anderswo stehen: «Die Ausrüstung schaltet bei Speisungsfehler automatisch auf die Sekundärspeisung um, um einen Konflikt auszulösen.»

Hintergrundinformation

Das TMN-Konzept beschreibt Komponenten zur Steuerung und Überwachung von Telecom-Netzen und wurde in ITU-T entwickelt. Die Komponenten werden via standardisierte Q3- oder X-Schnittstellen verbunden. Eine standardisierte Schnittstellen-Beschreibung sagt aus, welche Information zur Verfügung gestellt werden muss, was sie bedeutet und wie darauf zugegriffen werden kann. Bislang wurde dazu das CMIP-Protokoll auf einem OSI-Stack eingesetzt, seit kurzem auch CORBA und EDI auf TCP/IP-Stacks.

In TMN gibt es zwei Arten von Komponenten, *Agenten* und *Manager*. Manager steuern ein System und bieten Schnittstellen zum Anwender. Agenten nehmen die Managementbefehle des Managers entgegen und setzen sie in Netzelementbefehle um oder empfangen Mitteilungen vom Netzelement und leiten sie in geeigneter Weise an den Manager weiter.

Die MIB ist die Datenbank des Agenten, welche die Netzelement-Informationen gespeichert hält und auf die via Q3-Schnittstelle zugegriffen werden kann.

Scheme ist eine LISP-ähnliche Sprache. Sie zeichnet sich durch eine sehr kompakte Struktur aus und lässt sich deshalb leicht erlernen. Herausragendes Merkmal ist die Schreibweise: während bekannte Sprachen wie C++ eine Infix-Notation verwenden (z. B. 2+3), verwendet Scheme eine Prefix-Schreibweise (+ 2 3), was zunächst gewöhnungsbedürftig ist.

Abkürzungen

API	Application Programming Interface
CMIP	Common Management Information Protocol
CORBA	Common Object Request Broker Architecture
FAT	Factory Acceptance Tests
GDMO	Guidelines to the Definition of Managed Objects
GRM	General Relationship Model
IIOIP	Internet Interoperable Protocol
ITU-T	International Telecommunications Union – Telecom
MIB	Management Information Base
ODP	Open Distributed Processing
OMG	Object Management Group
RBF	Relationship Behaviour Formalisation
SDH	Synchronous Digital Hierarchy
TIMS	TMN-Information Model Simulator
TMN	Telecommunications Management Network
TTCN	Tabular Test Conformance Notation
UML	Unified Modelling Language

Bis anhin wurde das Verhalten des Systems jeweils einem Objekt zugewiesen (das Objektverhalten). Sind jedoch mehrere Objekte in ein Verhalten involviert, stellt sich die Frage, wer nun das Verhalten der anderen Objekte steuern soll. Diese Antwort hängt von der Implementierung des Herstellers ab. Der von TIMS gewählte Ansatz namens Relationship Behaviour Formalisation (RBF) [4] ordnet das Verhalten der Relation zu, die die involvierten Objekte bindet. Dadurch entfällt das oben erwähnte Zuordnungsproblem, und der Zugriff auf eine Objektinstanz vereinfacht sich, weil er unabhängig von Low-level-Implementierungen (z. B. Pointer) wird. Neues Verhalten kann schrittweise dem System hinzugefügt werden, was dem Prototyping-Gedanken sehr entgegenkommt. Verhalten wird immer durch ein Ereignis

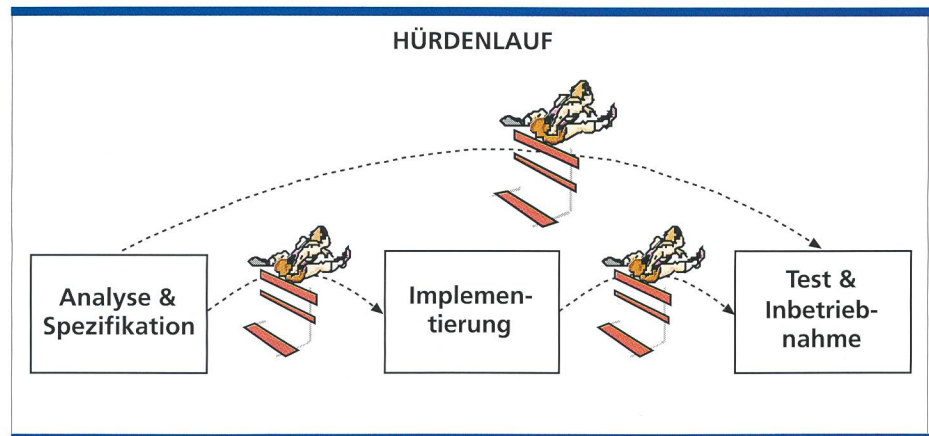


Bild 2. Der Hürdenlauf von Phase zu Phase lässt sich durch verbesserte Spezifikationen reduzieren.

ausgelöst. Als Auslöser dient z. B. eine Meldung, die etwas im System verändern will, beispielsweise einen Attributwert modifizieren, ein Objekt löschen oder eine Relation erstellen. Um sicherzustellen, dass keine unerwünschten Effekte auftreten, werden erwarteter Anfangs- und Endzustand (Pre- und Postcondition) für jedes Verhalten definiert. Pre und Post sind für die Validierung der Spezifikation unumgänglich. Auch bei der Verwendung als Beschaffungsgrundlage sind diese Randbedingungen wichtig, denn sie beschreiben das Verhalten des Systems, ohne den Algorithmus festzulegen (womit die Hersteller sicher nicht einverstanden wären).

TIMS im Einsatz

In mehreren Fallstudien wurde die Verwendung von TIMS für die Modellierung von Netzelement-Modellen (SDH Protection Switch, V5.1) und Netzmodellen (Anschlussnetz, Metran Xcoop) überprüft. Die meisten Schwierigkeiten bereiteten die Einarbeitung in das Fachgebiet, die klare Definition des TMN-Problems sowie die saubere Interpretation der Verhaltensregeln. Hier erwiesen sich Pre- und Postconditions als sehr nützlich, denn immer wieder übersahen die Entwickler vorher gemachte Annahmen [4]. TIMS eignet sich gut zur schrittweisen Entwicklung des Verhaltens von Objektmodellen, welche vorgängig mittels einer Software-Engineering-Methode (z. B. UML) beschrieben worden sind. Die Relationen, die Randbedingungen sowie die Möglichkeit, das Verhalten zu validieren, bieten dabei einen wesentlichen Vorteil gegenüber klassischen objekt-orientierten Frameworks (z. B. Smalltalk), um ein-

deutige Anforderungen an Hersteller abzuleiten.

Im Gegensatz zu professionellen Werkzeugen wie HP Openview lässt sich TIMS wesentlich rascher erlernen (<1 Monat), und rasch lassen sich damit Modelle realisieren (2–4 Wochen für eine vollständige Q3-Schnittstelle). Nebst einer sehr detaillierten Schnittstellen-Spezifikation können in TIMS auch ganze Managementprozesse und Szenarien erarbeitet werden, wie sie zuweilen zwischen Manager und Agent vorkommen. Die Verbindung mehrerer TIMS-Systeme erlaubt die Konstruktion ganzer TMN-Architekturen.

TIMS findet dort seinen besten Einsatz, wo der Anwender ein Bedürfnis nach

Referenzen

- [1] R. Sellin, TMN – die Basis für das Telekom-Management der Zukunft, R. von Deckers Verlag, Heidelberg 1995, ISBN 3-7685-4294-7.
- [2] ITU-T X.725 Generic Relationship Model, 1994.
- [3] ITU-T X.722 Guidelines to the Definition of Managed Objects, 1992.
- [4] R. Eberhardt, S. Mazziotta, D. Sidou, Design and Testing of Information Models in a Virtual Environment, Proceedings of the Integrated Management Symposium 1997 (IM'97), 1997.

wasserdichten Spezifikationen sieht (z.B. bei heterogenen TMN-Systemen oder für Standards) und wenn detailliertes Wissen über die Funktionsweise der Schnittstelle notwendig wird (z. B. für spätere Integrationstests). Als Entwicklungsplattform eignet sich TIMS aus Performance-Gründen weniger.

Ausblick

TIMS besteht aus zwei Komponenten: TIMS-Validator und TIMS-Emulator. Der Validator enthält alle Funktionalitäten für das Prototyping und Validieren von Informationsmodellen. Der TIMS-Emulator ergänzt das Paket durch die Schnittstellen-Möglichkeiten Q3 und CORBA. Nähere Informationen zu TIMS sind unter <http://www.vppt.ch/~tims> zu finden. In ersten Experimenten wurde die automatische Erzeugung von TTCN-Testsuites geprüft. TTCN eignet sich zur Beschreibung von Factory Acceptance Tests (FAT). Diese Anwendung könnte sich als sehr interessant erweisen, da die FAT-Entwicklung insbesondere von Szenarien als aufwendig gilt. Die Weiterentwicklung von TIMS in diese Richtung wird zurzeit geprüft. 11



Rolf Eberhardt, dipl. Informatik-Ing. ETH, trat nach seinem Studium Swisscom Corporate Technology bei. Er arbeitet in den Gebieten Anschlussnetz- und Servicemanagement mit dem Fokus auf End-zu-End-Prozesse, also alles vom Geschäftsvorfall bis hin zur Q3-Schnittstelle. Weiter vertritt er Swisscom in ETSI sowie beim Network Management Forum (NMF) und leitete mehrere Unisource OTC-Projekte. Als Programme Manager zeichnet er seit 1997 für das Programm «Operational Processes & Customer Care» verantwortlich.

Summary

TIMS – a pocket-size TMN laboratory

Is it possible to lower the overall introduction costs of a TMN system and to shorten the elaborate process from specification to the test by investing more into the specification phase? Presumably yes. A part of the answer can be found in the TMN information model simulator, TIMS, developed by Corporate Technology and Institut Eurécom. In TIMS the designer can develop executable TMN agents and managers in a short time and check them for internal conflicts. Although it was originally intended for use in the TMN area, any object simulation can be built. For this reason TIMS contains not only Q3 components but also components for COBRA interface specification. But TIMS provides also investment protection because the results of the prototype development can be used as the basis for procuring and creating test suites.

FORSCHUNG + ENTWICKLUNG

Sharp entwickelt neue Siliziumtechnologie für seine Flachbildschirme

Dünnschichttransistor-Displays bieten dem Anwender derzeit die besten und energiesparendsten Flachdisplays. Nachteilig ist der relativ hohe Aufwand für die Realisierung der Dünnschichttransistoren: Man darf davon ausgehen, dass der Anteil ihrer Fertigungskosten am Gesamtdisplay über 50% liegt. Sharp und das Semiconductor Energy Laboratory in Atsugi haben nun einen neuartigen «Continuous Grain Silicon» (CGS) Halbleiter entwickelt, dessen Elektronenmobilität an die der Silizium-Einkristalle herankommt. Genannt wird eine Elektronenmobilität von 300 cm²/Vs, was rund 600 mal so hoch ist wie in amorphem Silizium und noch immer viermal so hoch wie in Polysilizium. Damit kann man eine chipdünne Displaysteuerung bauen, sozusagen einen «Flachcomputer auf dem

Wafer». Bei Sharp wurden in dieser neuen Technik Prototypen von CGS-TFT-LCD aufgebaut. Mit drei solchen Elementen hat man die TFT-Treiber eines 60-Zoll-Projektionsbildschirms ausgerüstet: Er lief bei 13,8 MHz und lieferte hochauflösende Fernsehbilder.

Sharp Corporation,
22-22 Nagaike-cho
Abeno-ku, Osaka 545, Japan
Tel. +81-6-621 1221
Fax +81-6-628 1653
Tx J63428 LABOMET

Kleinster Transistor der Welt nutzt 40-nm-Technologie

Die Forschungslabors von Fujitsu haben in Zusammenarbeit mit der Kyoto-Universität den bisher kleinsten Transistor der Welt entwickelt, der nach dem Jahr 2012 in 256-Gbit-DRAM verwendet werden könnte. Der verwendete 0,04-µm-CMOS-Prozessor liegt in den Abmessungen etwa um den Faktor 10 unter dem, was heute in der Serienfertigung welt-

weit genutzt wird. Der Oxidfilm des Transistor-Gates beträgt nur noch 2 nm. Die Strukturen des Transistors wurden mit Elektronenstrahl geschrieben und durch anschließendes Ionenstrahlätzen realisiert. Für die Ionenimplantation wurde ein eigenes Verfahren entwickelt, das bereits in den nächsten Jahren fertigungsreif gemacht werden soll. Es erlaubt den Einschuss einzelner Bor-Ionen an gezielt ausgewählten Stellen für die Herstellung der «Source»- und «Drain»-Elektroden.

Fujitsu Limited
Marunouchi Center Building
6-1 Marunouchi 1-Chome
Chiyoda-ku, Tokyo 100, Japan
Tel. +81-3-3216-3211
Fax +81-3-3213-7174

Mitsubishi gibt Chipproduktion in den USA auf

Nachdem die amerikanische Halbleitertochter von Mitsubishi im letzten Jahr einen Verlust von mehr als 300 Mio. US-\$ gemacht hat, zieht Mitsubishi die Not-