2. Workshop Automatisierungstechnische Verfahren für die Medizin vom 25. bis 26. Feb. 1999 in Darmstadt

„Sicherheitstechnische Aspekte verteilter Informationssysteme in einem High- Tech-OP“

M. Stien, T. Lueth, A. Hein
Klinik für Mund-, Kiefer- und Gesichtschirurgie, Fachgebiet Navigation und Robotik,Charité, Berlin, Deutschland
E-Mail:malte.stien@ieee.org

ISBN: 318318317x
Pages: 13-14

Sicherheitstechnische Aspekte verteilter Informationssysteme in einem High-Tech-OP

M. Stien, T. Lueth, A. Hein

Klinik für Mund-, Kiefer- und Gesichtschirurgie
Fachgebiet Navigation und Robotik
Prof. Dr. Tim C. Lüth
Charité - Campus Virchow-Klinikum • Augustenburger Platz 1 • 13353 Berlin
malte.stien@ieee.org

Einleitung

In Operationssälen werden zunehmend komplexe informationstechnische z.T. semiautonome Geräte eingesetzt. Zu den sicherheitsrelevanten vernetzten und funktional integrierten Geräten in einem modernen OP gehören beispielsweise Beatmungsgeräte, Anästhesiegeräte, bildgebende Systeme, Patientenpositioniersysteme, Positionsmeßsysteme, Robotersysteme, Touch-Screen-Konsolen und aktive chirurgische Instrumente. Die Verwendung der Kombination der einzelnen Geräte zur Diagnose und Behandlung der Patienten kann als technischer Prozeß betrachtet werden. Die Störung oder der Ausfall einzelner Geräte oder Teile des Kommunikationsnetzwerkes muß mit minimalen Risiko für Patienten und medizinisches Personal verbunden sein.

Stand der Technik

Den Autoren ist gegenwärtig kein System bekannt, daß als Bibliothek zur Erstellung von sicheren Echtzeitverbindungen für integrierte OP-Anwendungen publiziert wurde. Generelle Konzepte zur Interprozeßkommunikation finden sich in [1, 2, 3]. Insbesondere unter dem Begriff *Middleware* werden viele Ansätze zu dieser Thematik zusammengefaßt [4].

Verwaltung der Prozeßzustände

Für diese Aufgabenstellung wird ein neues leistungsfähiges System zur sicheren und verteilten Repräsentation von Prozeßzuständen mit einem geeigneten Interaktionsprotokoll für den Prozeßzustandsaustausch zwischen den Geräten vorgestellt.

Jedes Gerät führt eine *lokale Prozeßzustandstabelle* (local–state–table, LST) der von ihm erfaßten, ermittelten bzw. kontrollierten (gesteuert bzw. geregelten) *lokalen Prozeßzuständen* mit *lokalem Prozeßzustandsnamen* (local–state–name, LSN) und *lokalem Prozeßzustandswert* (local–state–value, LSV). Die lokalen Prozeßzustände können auf rein geräteinternen Zuständen aber auch auf patientenbezogenen Zuständen oder auf Interaktionszuständen zwischen Gerät und Patient beruhen.

Die Änderung eines lokalen Prozeßzustands oder die erneute Erfassung wird zusammen mit der *lokalen Änderungszeit* (local–state–update–time, LSU) in der lokalen Prozeßzustandstabelle aktualisiert und gleichzeitig lokal in eine *lokale Protokolldatei* (local–protocol–file, LPF) geschrieben.

```
Local-state-table:
   <LSN>: <LSV> <LSU>,
   ...
```

Auf diese Weise ist der globale Prozeßzustand des Gesamtsystems zu einem beliebigen Zeitpunkt durch die Vereinigungsmenge der aktuellen lokalen Prozeßzustände gegeben. Er wird nicht zentral abgelegt oder repräsentiert.

Einige Geräte benötigen für den sicheren Betrieb möglichst aktuelle Kopien ausgewählter lokaler Prozeßzustände anderer Geräte. So ist es beispielsweise für einen Chirurgieroboter wichtig zu berücksichtigen, ob der OP-Tisch verstellt wurde. Jedes Gerät führt daher lokal eine *Zustandslieferantentabelle* (state–server–table, SST) mit den erforderlichen *externen Prozeßzustandsnamen* (external–state–name, ESN) und den Namen der *prozeßzustandsliefernden Geräte* (serving–device–name, SDN):

```
state-server-table:
   <ESN>: <SDN>,
   ...
```

Die externen Prozeßzustandsnamen müssen mit den lokalen Prozeßzustandsnamen der prozeßzustandsliefernden Geräte übereinstimmen. Aus der Kombination der Namen des prozeßzustandsliefernden Gerätes und dem externen Prozeßzustandsnamen setzt sich der entsprechende lokale Prozeßzustandsname zusammen:

```
<LSN> = <SDN>/<ESN>
```

Durch diese Art der Darstellung bleibt die ursprüngliche Quelle der Prozeßzustandsermittlung immer erhalten. Die Zustandslieferantentabelle muß nicht konstant bleiben bzw. es kann auch mehrere unterschiedliche Zustandslieferantentabellen geben, die durch einen *Lieferantentabellennamen* (server–table–name, STN) unterschieden werden können.

Welche Zustandslieferantentabelle gültig ist, kann von sogenannten *lokalen sicheren Prozeßzuständen* (local–safe–state, LSS) abhängen, die durch einen *Sicherheitszustandsnamen* (safe–state–name, SSN) und einen dazugehörigen *sicheren Zustandswert* (safe–state–value, SSV) definiert sind.

```
<SSN> <SSV>: <STN>
```

Die sicheren Gerätezustände sind selbstverständlich auch „normale“ lokale Prozeßzustände.

Prozeßzustände werden über ein Kommunikationsnetzwerk ausgetauscht. In einem ersten Realisierungsschritt wird der Austausch von Prozeßzuständen über ein Netzwerk, das "Broadcasting"

unterstützt, verfolgt. Die Geräte verwenden dabei ein *Sendeformat* bei dem der *Nachricht* (send–message–body, SMB) ein *Sendenachrichtenkopf* (send–message–head, SMH) vorangestellt wird.

```
Send message format: <SMH> <SMB>
```

Problemstellung

Um die Sicherheit des verteilten Gesamtsystems so hoch wie möglich zu halten, muß auf folgende Probleme eingegangen werden:

1 Das Gesamtsystem wird in beliebiger unbekannter Reihenfolge gestartet. Dabei müssen die einzelnen Geräte ihren lokalen Prozeßzustand aus dem globalen Prozeßzustand ermitteln können. Dabei kann es passieren, daß Geräte beim Gerätestart warten müssen, bis andere Geräte ihren lokalen Prozeßzustand ermittelt haben und diesen zur Verfügung stellen.

2 Ein Gerät fällt während des Betriebes unbeabsichtigt aus und muß nach einem Neustart wieder seinen lokalen Prozeßzustand ermitteln bzw. seinen vorherigen Prozeßzustand herstellen. In der Zwischenzeit müssen die anderen Geräte über den Geräteausfall informiert werden und ihren eigenen lokalen Prozeßzustand entsprechend korrigieren. Dies kann einen kompletten Neustart des Geräts, den Sprung zu einem sicheren Gerätezustand oder den Aufbau eines sicheren Gerätezustands nach einem Neustart bedeuten. Tatsächlich ist dies mit einem mehrstufigen Geräte- und Systemstart vergleichbar und wird daher im ersten Schritt nicht weiter betrachtet.

Um die Belastung des Kommunikationsnetzwerkes so gering wie möglich zu halten, muß bei dem Austausch der Prozeßzustände auf folgende Probleme eingegangen werden:

3 Prozeßzustände, deren Änderungen regelmäßig auftreten, müssen nur mit der maximalen Verarbeitungsfrequenz anderer Geräte nach außen übermittelt werden.

4 Prozeßzustände, deren Änderungen unregelmäßig auftreten, müssen sowohl auf Anfrage (z.B. bei Geräteneustart) als auch automatisch bei einer Veränderung an andere Geräte gesendet werden können.

5 Umfangreiche Prozeßzustandsänderungen, deren Verarbeitung durch andere Geräte unregelmäßig ist, sollen das Netz nicht unnötig belasten.

Methoden

Es werden verschiedene Typen von Sendungen im Sendenachrichtenkopf definiert. Sendungen vom Typ 1 stellen eine Änderungsmitteilung dar. Hierbei wird kein Zustandswert übertragen. In Meldungen vom Typ 2 wird ein Wert direkt übertragen. Sendungen mit einem kleineren Typwert werden priorisiert behandelt. Sendungen vom Typ 0 genießen maximale Priorität und werden daher für Notfallmeldungen verwendet.

Beim Start des Systems tauschen die Geräte ihre Zustandslieferantentabellen gegenseitig aus. In einem ersten Schritt wird ein einstufiger Gerätestart angenommen. Auf Verklemmungen muß daher beim Entwurf explizit geachtet werden. In einem lauffähigen System kann die Verklemmungsgefahr automatisch analysiert und auf die Notwendigkeit derer Beseitigung hingewiesen werden.

Jedes Gerät muß mit einer definierten Mindestfrequenz Nachrichten versenden. Sollten keine zu verschickenden Nachrichten anliegen, werden alive–Nachrichten verschickt. Werden von einem Gerät über eine sich aus der Mindestfrequenz ergebenden Zeit keine Nachrichten empfangen, so wird angenommen, dieses Gerät sei ausgefallen oder ausgeschaltet worden. Die Zustandskopien von diesem Gerät werden in allen anderen Geräten daraufhin als ungültig gekennzeichnet.

Jedes Gerät führt in der *Zustandsabonnementtabelle* (client–state–table, CST) eine Liste der aktuell abonnierten Zustände mit:

```
client-state-table:
   <State Name> <Client> <Frequency>,
   ...
```

Der jeweilige Zustand wird mit der aktuell maximal geforderten Frequenz gebroadcasted. Möchte ein Client Zustände eines Servers abonnieren, verschickt er unter Angabe des Zustandsnamens und der gewünschten Verarbeitungsfrequenz eine Nachricht an den Server. Daraufhin trägt dieser die Informationen in seine Zustandsabonnementtabelle ein.

Wird dabei als Verarbeitungsfrequenz der Wert Null angegeben, so wird der Wert vom Client unregelmäßig angefordert. Er muß nur über Änderungen informiert werden. Der Zustand selbst wird erst auf Anfrage hin übermittelt. Jeder Client muß aber Zustände die gebroadcasted werden auch annehmen, wenn sie mit einer höheren Frequenz als der von ihm geforderten versendet werden. Natürlich können über eine Leseanfrage des Clients alle Zustandswerte auch direkt vom Server abgefragt werden.

Zusammenfassung

In diesem Artikel wurden einige Forderungen an ein System durch konsistenten Verteilung von Prozeßzuständen aufgestellt, und ein Konzept vorgestellt, das diese Forderungen weitgehend erfüllt. Eine C–Software–Bibliothek wird in Kürze verfügbar sein.

Literatur

[1] Lüth, Tim (1998): Technische Multi–Agenten–Systeme, Hanser Verlag.

[2] Wettstein, H. (1993): Systemarchitektur, Hanser-Verlag.

[3] Mühlhäuser, M.; Schill, A.: Software-Engeneering für verteilte Anwendungen, Springer-Verlag, 1992.

[4] Tretsch, M.: Middleware: Schlüsseltechnologie zur Entwicklung verteilter Informationssysteme. Informatik Spektrum, 19 (5) (1996), pp. 249-256.