

~~Was hat Software-Engineering mit Brücken zu tun?~~

~~Wenn Applikationen nicht so tun, wie sie sollten ...~~

~~Die Object Engineering GmbH~~

~~Wettbewerb~~

~~Editorial: Info Bridge~~

~~Info Bridge kostenlos abonnieren / Feedback~~

Was hat Software-Engineering mit Brücken zu tun?



Mit der ersten Ausgabe der Kundenzeitschrift **Info Bridge** wollen wir Sie auf dem Laufenden halten, was im "stillen" Kämmerlein so alles läuft. Da die Object Engineering sich in den letzten beiden Jahren intensiv auf die Projektstätigkeit konzentriert hat, ist es gut möglich, dass sie nicht sehr viel von uns gehört haben. Das wird sich jetzt und in Zukunft auf eine möglichst vielseitige, aber vielleicht nicht immer so konforme Art ändern.

Vielleicht haben Sie sich bereits gefragt, wieso wir die Brücke zum Corporate Identity der Object Engineering gemacht haben? Denn, nebenbei bemerkt, seit die Brücke auf unserer Website erschienen ist, haben wir auch schon Werbung für Hoch- und Tiefbau-Artikel erhalten ... Das Positioning mit der Brücke kam zustande, als unser Werber und ich am Prospekt "Brückenbauen in der Landschaft der Objekte und Systeme" arbeiteten.

Per Definition heisst Engineering: "Planung, Design, Konstruktion oder Management von Maschinerie, Strassen, Brücken, Gebäuden, Wasserwegen usw." Die Herkunft des Wortes stammt aus dem Altfranzösischen engin = Fertigkeit und dies wiederum stammt vom Lateinischen ingenium = Erfindungskraft, Genie, Genialität. Und was symbolisiert "Engineering" in der Realität am Besten? Sicher vieles, aber wenn Sie mich fragen (und das hat damals unser Werber gemacht), dann würde ich antworten "Brückenbau".



Engineering ist mit unserer Tätigkeit verbunden, weil ich ursprünglich als

Elektroingenieur und später als Informatiker dort mein Herz verloren oder angesiedelt habe. Es ist eine Sache des Seins, man ist Ingenieur oder eben nicht. Und wenn man es ist, dann ist das Herz voll dabei. Software-Entwicklung und Engineering ist noch nicht überall ein vertrautes Paar, obwohl das Verantwortungsbewusstsein eines Ingenieurs dort mehr als am Platz ist, ja eigentlich hingehört. Die Object Engineering hat sich vorgenommen, Software-Engineering zu fördern und es auch zu leben. Produktivität - aber vor allem Qualität und Langlebigkeit (ein Fremdwort in unserer Branche?) können nur mit einem systematischen Entwicklungsprozess, normierten Komponenten und "Bauteilen", Planung, Strategien, Weitsichtigkeit - aber auch Pragmatik - erreicht werden. Als Unternehmen mit kleinem Personenaufwand gehört Obiges erst recht zum Alltag, wenn man morgen noch auf dem Markt sein will. Die Disziplin des Brückenbaus hat noch Aspekte, die auch bei einem guten Software-Design, wenn auch nicht immer offensichtlich, zu finden sind: Ästhetik, Einpassung (resp. Anpassung) in die Umwelt, Zweckerfüllung und im Dienste der Menschen zu stehen. Unterdessen sind Brücken fast zu meinem Hobby geworden und so ist es richtig, weil mein Beruf auch mein Hobby ist.

Ihr Echo auf unsere **Info Bridge** interessiert uns natürlich brennend und wir bitten Sie, uns Ihre Meinung, Kritik, Anregungen etc. mitzuteilen. Mit Internet dem Kontaktformular geht dies unkompliziert und schnell. Ich hoffe, wir können mit der **Info Bridge** eine Brücke zu Ihnen schlagen und wünschen Ihnen viel Spass beim Lesen.

Anares Koch, Object Engineering GmbH

Wenn Applikationen nicht so tun, wie sie sollten ...



... oder: Wenn verteilte Applikationen das Weite suchen ...

Client/Server-Systeme und überhaupt jede Art von verteilten Applikationen haben ihren Vorteil vor allem darin, die nötige Flexibilität der IT-Landschaft bei den gelegentlich sich ändernden Geschäftsprozessen eines Unternehmens zu gewährleisten. Oft wird hauptsächlich über Tools und Technologien der Entwicklung debattiert und dabei werden die Bedürfnisse des Betriebes vernachlässigt. Wenn dann eine der System-Komponenten aber nicht funktioniert, richtet sich die ganze Aufmerksamkeit bis ins obere Management auf die Betriebsmannschaft, die dann sozusagen mit gebundenen Händen und oft auch mit verbundenen Augen im Schussfeld stehen.

Betriebsunterbrüche verhindern

In vielen Fällen könnten Betriebsunterbrüche jedoch vorausgesehen werden, wenn man die richtigen Hilfsmittel dazu hätte. Bei einem Auto prüft man periodisch (oder sollte wenigstens) Ölstand, Reifendruck, Kühlwasserstand usw.; bei Maschinen führt man eine regelmässige Wartung mit Messungen durch, bei Grosscomputern überwacht man laufend die Ressourcen-Auslastung usw. Mittels solch einfacher Diagnosen kann oft im frühen Stadium etwas unternommen werden, bevor das ganze System zusammenbricht. Aber was ist zu bewerkstelligen, wenn es sich um eine abnutzungsfreie aber dafür in der Regel nicht perfekte Applikations-Software handelt, deren Komponenten zudem noch "frei" im Netz auf verschiedenen Rechnern verteilt sind?

Typischerweise sind solche Komponenten durch eine sogenannte Middleware wie CORBA, RMI/EJB, DCOM u.a. miteinander gekoppelt. Auf jede Komponente kann über eine Schnittstelle, die sie nach aussen im Netz repräsentiert, zugegriffen werden. Eine Schnittstelle ist eine Sammlung von Funktionen mit Parametern, welche über die Middleware aktiviert werden können. Solche Komponenten können mit Inseln im Ozean verglichen werden, wobei das Meer die Middleware, die Häfen der Inseln die Schnittstellen und die dazwischen kursierenden Schiffe die Informations-Meldungen sind.

Diese Schnittstellen sind in der Regel applikationspezifisch zugeschnitten. Es ist praktisch ein fast unlösbares Problem, ein Diagnoseprogramm zu entwickeln, durch welches alle Applikationen überwacht und geprüft werden können; insbesondere darum, weil diese Applikationsschnittstellen und die dahinter liegenden Implementationen laufenden Änderungen der Applikations-Anforderungen unterworfen sind. Auch für das Betriebspersonal wäre dies nicht zu bewältigen, müssten sie doch alle Applikationen im Detail kennen, um diese richtig zu betreuen.

Es gibt aber eine einfache Lösungsvariante, die uns da weiterhilft. Wenn man z.B. eine auf CORBA basierte Middleware verwendet, können Applikationen mit mehreren, verschiedenen Schnittstellen realisiert werden. Die aussenstehenden Client-Applikationen können z.B. die Applikationsschnittstellen a1 von Applikation A verwenden, ohne davon betroffen zu sein, dass die Applikation A noch weitere Schnittstellen a2 und a3 hat (Abb. 1).

Dieser Umstand kann ausgenutzt werden, um jeder Applikation eine identische Diagnostik-Schnittstelle zu geben, welche genereller Natur ist. Nur ist es notwendig, dass eine Diagnostik-Client-Applikation gebaut werden muss, welche jede Applikation, die mit der genannten Diagnostik-Schnittstelle versehen ist, überwachen kann. Die Applikationen müssen zwar die Diagnostik-Funktionen implementiert haben und diese über die Diagnostik-Schnittstelle nach aussen repräsentieren. Wenn die Diagnostik-

Client-Applikation geschickt implementiert ist, kann sie jedoch gleichzeitig den Zustand mehrerer Applikationen überwachen.

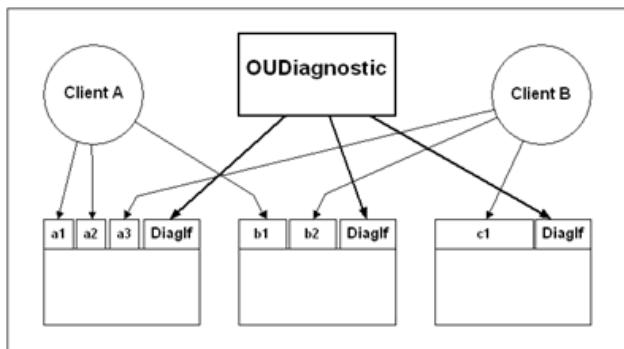


Abbildung 1: Konzept OUDiagnostic

Funktionen von Applikationen

Welche Funktionen müsste ein generisches Diagnostik-Interface einer Applikation unterstützen?

Unsere Erfahrung mit verschiedensten, verteilten Applikationen hat folgende Funktionalitäts-Anforderungen herauskristallisiert:

- aktuelle Version einer Applikation zur Laufzeit abfragen
- momentanen Zustand (Status) einer Applikation abfragen
- momentane Auslastung einer Applikation abfragen (in %) .
- Liste von applikationsspezifischen Diagnose-Funktionen ausgeben
- Auslösen von Diagnose-Funktionen der Applikation
- Verändern des Detaillierungsgrades für die Fehler-Trace-Ausgabe zum Suchen eines Fehlers

Das Abfragen der Liste verfügbarer Diagnostik-Funktionen ist der wichtige generische Teil, da jeder Service seine spezifischen Diagnostik-Routinen hat. Typischerweise melden diese Operationen lediglich detailliertere Informationen über den Zustand der Applikations-Komponente zurück. Dies können zum Beispiel Anzahl Client-Objekte, Anzahl offener Verbindungen zu anderen Services usw. mit detaillierten Angaben zu welchem Client diese gehören, wann sie eröffnet wurden und in welchem Zustand sich diese zur Zeit gerade befinden.

Eher selten lösen diese Diagnostik-Funktionen eine Zustandsveränderung im Server aus, und wenn dies der Fall ist, dann wäre die Verwendung auf eine bestimmte Benutzergruppe zu beschränken.

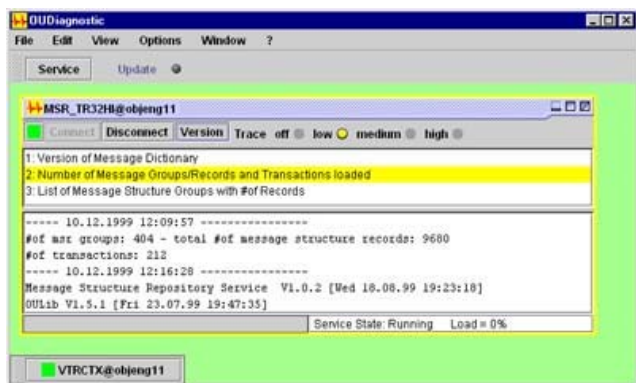


Abbildung 2: Benutzeroberfläche OUDiagnostic-Tool

Jede Funktion in der Liste kann über den Diagnostic-Client (vgl. Abbildung 2) ausgelöst werden und die Resultate werden dahin zurückgeliefert. Das heisst, dank der darunterliegenden Middleware kann der Diagnostik-Client jede im Netz zugreifbare Applikation überwachen, was dieses Prinzip sehr mächtig macht. Die zurückgelieferten Resultate sind in der Regel in lesbarem Text, wobei die Zeichensatzkonversion automatisch durch die darunterliegende Middleware gelöst wird.

Abfragen des aktuellen Status

Das Abfragen des aktuellen Status und der aktuellen Belastung erfolgt über die nicht generische Schnittstelle (in CORBA IDL definiert). Dies kann vom DiagnostikClient auch periodisch und automatisch gemacht werden, und aufgrund dessen die Farbe des zur Applikation zugehörigen Fensters entsprechend geändert werden (grün = ok), so dass die für den Betrieb verantwortliche Person auf den ersten Blick sieht, wenn ein Service überlastet, in einem kritischen Zustand (gelb) oder sogar unterbrochen (rot) ist.

Unsere Erfahrung mit diesem einfachen Werkzeug ist überraschend gut. Bereits in der Entwicklung der Applikations-Services, aber auch für unsere Kunden in der Testphase und später im produktiven Betrieb, hat sich dieses einfache Prinzip sehr bewährt.

Gerade die Flexibilität und die Einfachheit (für die Entwicklungsseite wie für den Betrieb) ist überzeugend und die Funktionalität kann laufend dem Bedarf angepasst und ausgebaut werden.

Wenn Sie Interesse an diesem Konzept gefunden haben, empfehlen wir Ihnen die verwendete IDL-Schnittstellen-Definition und die zugehörigen Code-Beispiele von unserer Home-Page zu holen und darauf aufzubauen. (Der URL ist: <http://www.objeng.ch/DE/page/software-diagnostik.html>) Die verfügbaren Teile können Sie unter Quellenangabe und Einhaltung des Urheberrechts kostenlos verwenden. Falls Sie die zugehörige in Java geschriebene Diagnostik-Client-Applikation möchten, senden Sie uns bitte eine per [Kontaktformular](#) und wir werden Ihnen das entsprechende Zip-File kostenlos zusenden. Weitere Fragen beantworten wir gerne über das [Kontaktformular](#) unter dem Stichwort: OUDiagnostic.

Die Object Engineering GmbH Zielsetzungen



Object Engineering GmbH ist ein Unternehmen mit qualifizierten Ingenieuren, die in praxisbewährten und neusten Technologien der Informatik-Industrie sattelfest sind und durch effiziente Anwendung und laufende Weiterbildung dies in die Unternehmen und Projekte der Kunden tragen und etablieren können.

Zufriedenheit

des Kunden mit den professionellen Leistungen der Object Engineering GmbH ist oberstes Gebot des Unternehmens.

Know-how + Background

- Ingenieure mit fundierter Informatik-Ausbildung (HTL, Universitätsabschluss)
- Entwicklung von integrierten Informationssystemen mit Client/Server-Technologie mit meldungsorientierter Methodik und Distributed Computing
- Entwicklung von Architekturen und Applikationen für objektorientierte und verteilte Systeme auf der Basis von CORBA^(TM) (Common Object Request Broker Architecture [OMG[®]])
- Migration von bestehenden Systemen in eine verteilte System-Umgebung
- Software-Engineering: Einführung von Methoden und OO-Case-Tools zur Software-Entwicklung
- Ausgewiesene Projekterfahrung mit objektorientierter Entwicklung mit C++ , Booch u.a.
- Mit 20 Jahren Erfahrung in System-Design und System-Programmierung
- Langjährige Erfahrung in Entwicklung von Interpretern und Compilern.

Dienstleistungen

Engineering

- Analyse, Design, Programmierung
- System-Architekturen für verteilte Systeme
- Migrations- und Integrations-Projekte
- Distributed Object Computing

Consulting/Coaching/Ausbildung

- Verteilte Systeme
- Objektorientierte Technologie
- Distributed Object Computing

Eingesetzte Methoden/Technologien/Tools

Objektorientierung

- Booch-Methode
- UML

Distributed Object Computing

- CORBA^(TM)-Technologie, EJB
- Orbix[®], u.a.

Betriebssysteme

- UNIX[®], AIX^(TM), HP-UX^(TM), Solaris^(TM), SCO, Linux
- Windows/NT^(TM) u.a.

Programmiersprachen

- C, C++, Java[®] u.a.

Die Brücke ist laut Brockhaus Enzyklopädie: *ein Ingenieurbauwerk zur Überführung eines Verkehrsweges, eines Kanals, von Wasser oder Gasleitungen und dergleichen über ein Hindernis*. Brücken begegnen uns tagtäglich auf dem Arbeitsweg, auf der Autobahn, in der Stadt, auf dem Land, im In- und im Ausland. Oft greifen wir zur Kamera und fotografieren die Brücke: die Tower Bridge in London, die Brooklyn Bridge in New York oder die Karlsbrücke in der Goldenen Stadt Prag.

Quizfrage: Das Foto zeigt die Brooklyn Bridge. Doch irgendetwas stimmt nicht.



Lösung

Dieses Bild ist seitenverkehrt - weil es uns so besser gefiel.

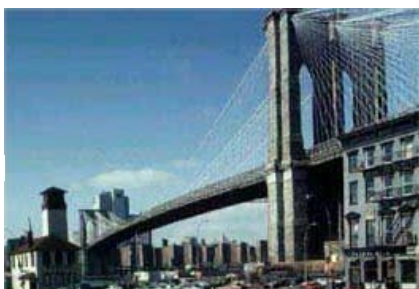


Dies ist das Bild, so wie es damals auch der Fotograf gesehen hat.

Und das fanden wir dann auch noch raus - die Brooklyn

Bridge ist die Brücke im Hintergrund, die Brücke im Vordergrund ist die Manhattan Bridge

Deshalb hier noch ein Bild der Brooklyn Bridge



Der Gewinner dieses Wettbewerbs ist **Herr Daniel Widler**, der mit der richtigen Lösung einen Eintritt ins Verkehrshaus Luzern für zwei Personen gewonnen hat.

Editorial

Liebe Interessentin
Lieber Interessent



Wir werden mit unserer **Info Bridge** auch einmal ein allgemeines, soziales Thema ansprechen. oder eine Prognose wagen. Der Innenteil ist dem für Technik interessierten Leser gewidmet. In der vorliegenden Nummer geht es um Diagnostik von verteilten Applikationen.

Da wir nichts davon halten, Know-how zurückzuhalten, sondern Erfahrungen gerne weitergeben, damit es anderen das Leben erleichtert, stellen wir Tipps und Tricks, auch einfache Techniken oder Hilfsmittel vor.

Die letzte Seite ist jeweils Informationen über unser Unternehmen und unsere Dienstleistungen gewidmet sowie einem lockeren Wettbewerb, mit dem wir Ihnen die Begeisterung und Faszination von Brücken vermitteln möchten.

Mit herzlichen Grüßen

Andres Koch
dipl. Ing. HTL,
M. Math
Geschäftsführer
Object Engineering GmbH