



## Open Ticket Request System

# Ablaufsicherung

Lukas Grunwald

Für Kundenkontakte oder auch den internen Service setzen viele große Firmen und Call-Center Ticketing-Systeme ein. Eine kostenlos verfügbare Software musste zeigen, ob diese Technik auch den Alltagsbetrieb einer mittelständischen Firma vereinfachen kann.

Jeder, der schon einmal eine technische Anfrage gestellt hat, kennt die automatisch von Ticketing-Systemen (auch Trouble-Ticket-Systeme – TTS – genannt) vergebenen Bearbeitungsnummern, ohne die sich kein Vorgang zu Ende bringen lässt. Solche Systeme sind meistens komplex und häufig teuer, doch es gibt auch kosten-

lose TTS wie das Open Ticket Request System, eine GPL-Software.

OTRS ist in Perl geschrieben und läuft unter Linux, Solaris, AIX, FreeBSD, OpenBSD, Mac OS 10.x und Windows. Für eine gute Performance sollte auf dem Webserver das Modul Mod\_Perl eingebunden sein.

Nachdem die CPAN-Module und die Software installiert sind, kann der TTS-Administrator das System via Weboberfläche anpassen und den Workflow modellieren. Im Test diente MySQL als Backend-Datenbank, aber auch Oracle und Postgres lassen sich via Perl-Konfiguration einbinden. OTRS musste nicht nur Kundenanfragen, sondern auch Roboter-Mails von einem Internet-Domain-Registrar sowie Systemüberwachungsmeldungen einer Nagios-Installation verarbeiten.

## Automatische Eskalation

Ticket-Mails sind über eine POP3-Mailbox oder eine direkte Pipe wie bei

Procmail zustellbar. Mittels vorher definierter Schlüsselwörter lassen sich die eingehenden Aufträge für neue Tickets auf verschiedene Warteschlangen verteilen. Abhängig davon können Anfragen, die der „Agent“ – so die Bezeichnung des menschlichen Bedieners – nicht innerhalb einer bestimmten Frist bearbeitet, zur nächsthöheren Instanz gelangen.

Ein Agent kann sich je nach Berechtigungskonzept bei den Tickets bedienen und zum Abarbeiten in seine persönliche Warteschlange kopieren. Das Ticket ist dann für alle anderen Agenten gesperrt. Verschlampt ein Agent das Ticket durch Nichtbeachtung, lässt es sich wieder freigeben und einem neuen Bediener zuweisen – ein Wunschtraum jedes Telekom-Kunden.

Als besonderer „Bediener“ steht der Generic-Agent zur Verfügung – ein Perl-Modul, das Cron-gesteuert spezielle Aufgaben erfüllen kann, etwa das Löschen von Datenmüll wie Viren und Spam oder das Verschieben von Roboter-Mails in eine eigene Warteschlange. Ein Wermutstropfen: Solche Funktionen sind als Perl-Module zu programmieren, und wer das nicht beherrscht, steht hier im Regen.

## Dokumentationshilfe enthalten

OTRS kann den Kunden automatisch über jede Zustandsänderung eines Tickets informieren. Vom Eröffnen bis zum Schließen lässt sich jede Aktion lückenlos dokumentieren und später

### DATEN UND WERTUNG

#### OTRS 1.2

Open Ticket Request System

**Bezugsquelle:** www.otrs.org

**Systemvoraussetzung:** Perl

**Preis:** kostenlos, kommerzieller Support und Installations-CDs verfügbar über die OTRS GmbH (www.otrs.de), Version 1.2.4 für Windows mit Handbuch und 90 Tagen Support 999 €

- ⊕ GPL-Software
- ⊕ großer Funktionsumfang
- ⊕ einfach zu erweitern
- ⊕ Volltextsuche über alle Tickets
- ⊖ zum Teil Perl-Wissen notwendig

**Ganz ohne Programmierung lassen sich nicht alle Feinheiten nutzen (Abb. 1).**

nachvollziehen. Agenten können vorgefertigte Antwortvorlagen bekommen, sodass sich mit OTRS ein komplettes Call-Center betreiben lässt. Die Software ist vielseitig und könnte so mancher proprietären Lösung als Vorbild dienen. Ob proprietär oder nicht: Der Aufwand für die Abbildung der Organisation der Arbeitsabläufe ist bei allen TTS praktisch gleich. Aber wenn keine Lizenzkosten anfallen, kann sich dieser Einrichtungsaufwand bei einer mittleren Firma schon innerhalb eines Tages amortisieren.

Ein Ticketing-System kann mehr als nur Kundenanfragen bearbeiten. Nichts spricht dagegen, es etwa für das Mahnen von Kunden einzusetzen,

```
Kernel/Config/GenericAgent.pm - config file of generic agent
Copyright (C) 2002-2004 Martin Edenhofer <martin@codeotrs.org>
...
package Kernel::Config::GenericAgent;
use strict;
use vars qw($VERSION @ISA @EXPORT %Jobs);
require Exporter;
@ISA = qw(Exporter);
@EXPORT = qw(%Jobs);
$VERSION = '$Revision: 1.7 $';
$VERSION =~ s/^(\d+\.\d+)(.*)$/$1/;
...
# config options
# -----
%Jobs = (
    'delete all tickets with subject "SPAM" in queue hostmaster' => {
        Queue => 'Internet.Hostmaster',
        Subject => '%SPAM%',
        New => {
            Delete => 1,
        },
    },
    'delete all junk tickets queue junk' => {
        Queue => 'junk',
        New => {
            Delete => 1,
        },
    },
);
# end of config options
};
```

die ihre Rechnungen nicht termingerecht bezahlen oder für die Unterstützung interner Projektprozesse. Dank der Eskalations-Methode lässt sich konzentriert melden, wo es klemmt, ohne die höhere Instanz mit Informationen zu überfluten.

Die Dokumentationsfunktionen können dazu beitragen, dass die interne IT-Abteilung ohne nennenswerte Mehrkosten den Betrieb schnell und einfach

ISO-900x-konform gestaltet. Damit sich Aktionen nachvollziehen lassen, kann man via Volltextsuche oder mittels Filterfunktionen direkt nach einem Vorgang suchen und ihn – ebenfalls dokumentiert – prüfen.

## Fazit

OTRS ist ein „Must-Have“ für Firmen, die mit Kunden, Support oder Tickets zu tun haben, egal ob für Pre-Sales-Anfragen oder den Wartungsdienst. Durch die offene Struktur lassen sich CRM-Systeme oder so genannte Standardsoftware einfach via DBI (Database Interface) anfügen – Perl-Kenntnisse vorausgesetzt. Das ganze System ist mittels Perl recht einfach zu dokumentieren und hat sich im Praxistest als stabil erwiesen. (un)

LUKAS GRUNWALD

arbeitet als Consultant bei der DN Systems GmbH in Hildesheim und ist in diverse freie Softwareprojekte involviert.

