

ITIL as a Testbed for SOA

Michael Rulf

VP Advanced Engineering

Michael.Rulf@usi.com

Why is SOA useful for ITIL?



An AT&T Company

- **IT is (& has been) under increasing pressure to do more with less**
- **Sysadmin salaries are the most prominent line in IT budgets, but often executing mundane ad-hock tasks**
- **ITIL is all about process, repeatability & standardization**
- **There are existing toolkits out there providing prebuilt processes for ITIL "best practices"**
- **IT folks love scripts to simplify/automate repetitive tasks**
- **ITIL processes are not as organizationally challenging as larger end-to-end business processes**
- **Many ITIL processes form the basis of larger business-critical business processes**

Take a building block approach

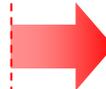


An AT&T Company

- **Service-enable small units of custom code**
 - Large amounts of Perl code & shell/VB scripts
- **Re-use these services as part of small process flows**
 - Approval processes
 - Account creation & maintenance
- **Smaller process flows become useful sub-processes of end-to-end business flows**
 - HR onboarding

BACKGROUND/CHALLENGE

- USInternetworking specializes in managed enterprise and eBusiness solutions and on-demand services for Fortune 1000 companies
- Well documented processes, but fragmented operations due to manual processes
- Many groups involved in each process
- Audit requirements – both internal and external



SOLUTION

- Used BPEL to orchestrate and manage multi-product process flow
- Created a set of “wrappers” that can be used to service-enabling scripts
- Utilized human workflow components to track manual tasks for audit compliance
- Reused legacy tools & scripts



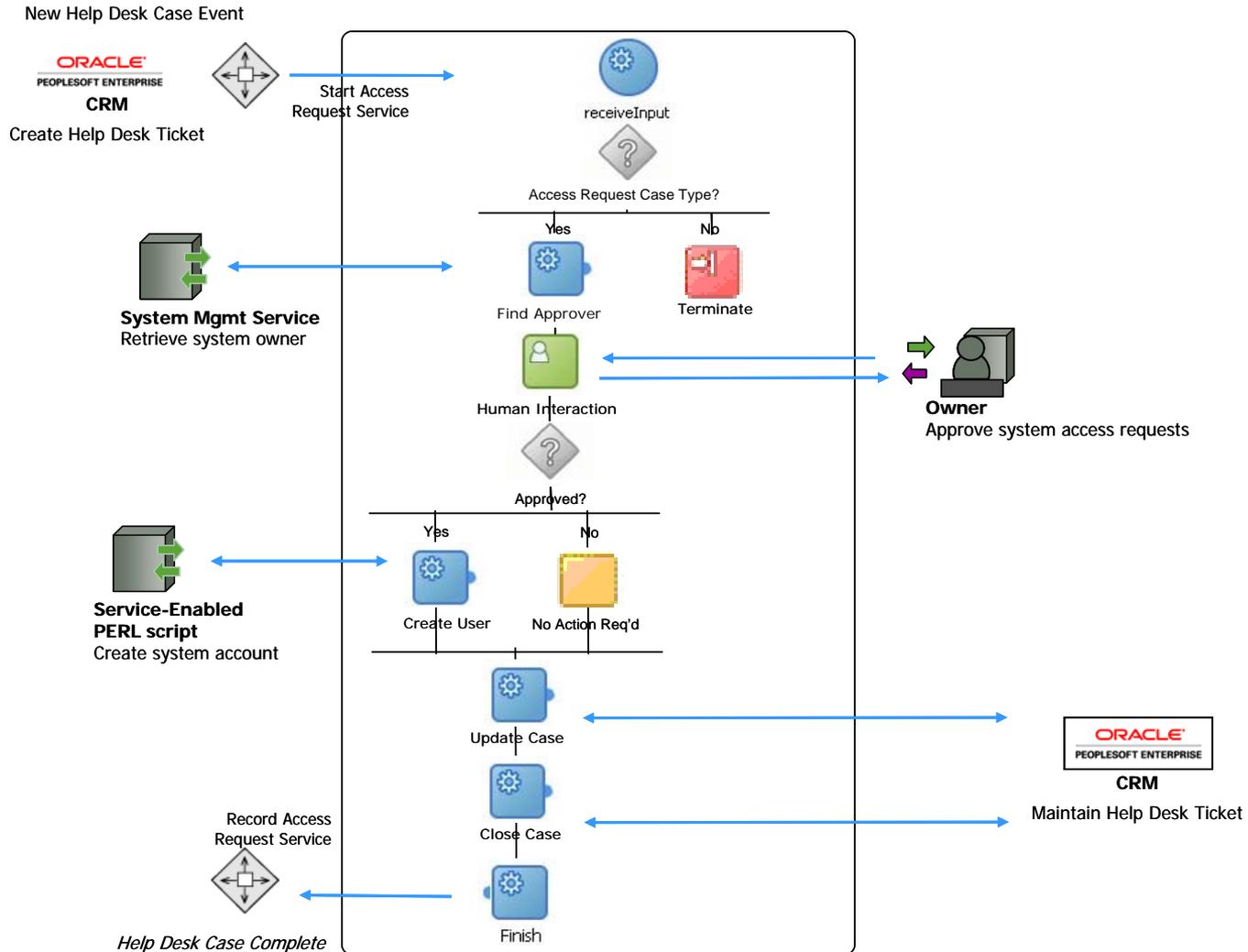
RESULTS

- Reduced labor costs & wall clock time per process by a minimum of 50%
- Reduced rework labor through automation of complex multi-task processes
- Increased compliance with audit requirements through automation and standards
- Developed a toolkit that is applicable for business processes outside of IT

ITIL Process Example: User Management



An AT&T Company





An AT&T Company

Let's see a demo...

Service-enabling a Perl Script - Legacy Perl Script



An AT&T Company

```
#!/usr/local/bin/perl
#
# Author: Quadir Kareemullah USi
#
# Purpose: UNIX account creation module
#
package UserAdd;
use strict;
use warnings;
use Data::Dumper;
my $groupadd_cmd = "groupadd";
my $useradd_cmd = "useradd";
my $def_homedir = "/home";
my $def_shell = "/bin/bash";
sub useradd {
    my ($fullname, $username) = shift;
    my $result = { error => 1, status => "Could not run useradd() function" };
    my $cmd = "$groupadd_cmd $username 2>&1";
    my $out = ` $cmd `;
    if($out =~ /exists/) {
        $result->{error} = 1;
        $result->{status} = "Could not add $username user, Group $username already exists.";
    }
    else {
        $cmd = qq($useradd_cmd -c "$fullname" -d $def_homedir/$username -s $def_shell -g $username $username 2>&1);
        $out = ` $cmd `;
        if($out =~ /exists/) {
            $result->{error} = 2;
            $result->{status} .= "Could not add $username user, User $username already exists.";
        }
        else {
            $result->{error} = 0;
            $result->{status} = "User $username successfully added.";
        }
    }
}
}
1;
```

Service-enabling a Perl Script - Wrapping Package



An AT&T Company

- Create a “wrapping” package that will expose UserAdd.pm via SOAP.
- The “use UserAdd” statement tells the new module to use our legacy code in UserAdd.pm.
- The two SOAP lines that follow are what enables the module to be called as a web service.
- A “sub” section then follows for each function in the original PERL module.
- A separate wrapping module allows you to add additional SOAP error handling to conform to a web services methodology as found in the “die” statement near the end of the script.
- **The original code is unchanged**

```
#!/usr/local/bin/perl
package UserAddWS;
use strict;
use warnings;
use diagnostics;
use Data::Dumper;
use lib './modules';
use User<service name="UserAddWS_Service">
  <documentation>WSDL File for UserAddWS</documentation>
  <port binding="tns:UserAddWS_Binding"
    name="UserAddWS_Port">
    <soap:address
      location="http://mdsxaaw05.usi.net/soap/UserAddWS.pl"/>
    </port>
  </service>
</definitions>
Add;
use SOAP::Transport::HTTP;
SOAP::Transport::HTTP::CGI
-> dispatch_to('UserAddWS')
-> handle;
sub useradd {
  my $self = shift;
  my $result = UserAdd::useradd(@_);
  if($result->{error}) {
    die SOAP::Fault->faultcode( $result->{error} )
      ->faultstring( $result->{status} );
  }
  return $result->{status};
}
1;
```

Service-enabling a Perl Script - WSDL Definition



An AT&T Company

- WSDL definition enables the new package to be called by a BPEL process or via statically typed language like Java or VB
- CPAN (<http://www.cpan.org>), provides PERL modules that help automate the creation of WSDL files.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="UserAddWS"
  targetNamespace="http://mdsxaaw05.usi.net/wsdl/UserAddWS.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://mdsxaaw05.usi.net/wsdl/UserAddWS.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="UserAddRequest">
    <part name="fullname" type="xsd:string"/>
    <part name="username" type="xsd:string"/>
  </message>
  <message name="UserAddResponse">
    <part name="status" type="xsd:string"/>
  </message>
  <portType name="UserAddWS_PortType">
    <operation name="useradd">
      <input message="tns:UserAddRequest"/>
      <output message="tns:UserAddResponse"/>
    </operation>
  </portType>
  <binding name="UserAddWS_Binding"
    type="tns:UserAddWS_PortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="useradd">
      <soap:operation soapAction=""/>

```

```
<input>
  <soap:body
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="urn:UserAddWS"
    use="encoded"/>
</input>
<output>
  <soap:body
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    namespace="urn:UserAddWS"
    use="encoded"/>
</output>
</operation>
</binding>
<service name="UserAddWS_Service">
  <documentation>WSDL File for UserAddWS</documentation>
  <port binding="tns:UserAddWS_Binding" name="UserAddWS_Port">
    <soap:address
      location="http://mdsxaaw05.usi.net/soap/UserAddWS.pl"/>
  </port>
</service>
</definitions>
```

Critical Success Factors:

- Start small and build reusable components
- ITIL is accepted as a proven method for reducing costs in IT
- Focus on the business requirements/value, not the software & technology capabilities
- Leverage self-service
- Organizational buy-in and processes – managing change critical

Future Compatibility:

- SOA and Web Services ensure compatibility with Oracle's Fusion strategy
- Separates “process” from “product” to streamline upgrades and future changes

How do customers get started with Fusion Middleware?



An AT&T Company

PeopleSoft customers:

- Upgrade to PeopleTools 8.48 or higher
- Start using Integration Broker for interfaces and extensions
- Investigate PeopleSoft Directory Interface (PDI)

EBusiness Suite customers:

- Upgrade to 11.5.10 at a minimum (ATG 11.5.10 Rollup 4)
- Tightly integrated with Fusion Middleware

All ERP customers:

- Fusion Identity Management
- Fusion SOA & BPEL



An AT&T Company

ITIL as a Testbed for SOA

Michael Rulf

VP Advanced Engineering

Michael.Rulf@usi.com