NAME

      ITSMTrace – trace links between services and config items

SYNOPSIS

      This OTRS package adds the capability to visualize relationships between services and ITSM configuration items.

      Relationships in OTRS are expressed by links that carry a certain semantic in their name (link type). A set of useful link types is provided with ITSM, e.g. `DependsOn` to indicate that a given service or config item (CI) depends on another CI. The zoom view of OTRS allows to see links going to/from a service or CI at that level, but not beyond. **ITSMTrace** can show *all* links in and out of the current service/CI and follow them recursively within settable constraints (specific link types, specific CI types, recursion depth, incident state, etc.).

      Graphical output is generated through the open-source package `graphviz` and can be accessed from within the web UI either for a single service/CI or for all of them. Automatic clustering helps to reduce the amount of information presented and improve the understandability of the generated images. Textual output is available through a command-line interface.

DESCRIPTION

      This section explains the basic concept behind the provided capabilities and some of the assumptions on which they rely. Please see the perldoc of `Kernel::System::ITSMTrace` for details about option names and accepted values.

      Data Model

      In the context for which **ITSMTrace** has initially been developed, there is the need to model technical IT infrastructure with lots of applications. A "service" as exposed in OTRS typically has one or more associated CIs of type `Technical Service` on which it depends. The newly defined CI type `Technical Service` is usually a server/service process, something that "runs" and can be started/stopped and itself depends on other CIs of type `TechnicalService`, `Computer`, `Data` (not in the default ITSM catalogue), or `Software`. The granularity and extent of modeling is up to the user of this software.

      Example:

```
% bin/otrs.ITSMTrace.pl –CompactTrace –ServiceID 222
OTRS [Service#222]
 -> <<DependsOn>>  OTRS for IS/IT [ITSMConfigItem::TechnicalService#103]
    -> <<DependsOn>>  ITSM 3.0.4 [ITSMConfigItem::Software#99]
    -> <<DependsOn>>  OTRS 3.0.9 [ITSMConfigItem::Software#98]
    -> <<DependsOn>>  MySQL Database "otrs3" [ITSMConfigItem::Data#97]
      -> <<DependsOn>>  MySQL Server [ITSMConfigItem::TechnicalService#100]
        -> <<DependsOn>>  atvievl5b364 [ITSMConfigItem::Computer#102]
    -> <<DependsOn>>  Apache httpd [ITSMConfigItem::TechnicalService#101]
      -> <<DependsOn>>  atvievl5b364 [ITSMConfigItem::Computer#102] (already visite
```

      Object– and Link-Types

      ITSMTrace wraps services and CIs as "objects" with a common interface, their object types being `Service`, `ITSMConfigItem` and all CI subclasses, e.g. `ITSMConfigItem::Computer`. These object types can be used to restrict the objects that are candidates to be shown on a trace.

      Recognized link types are those configured into OTRS+ITSM (`DependsOn`, `AlternativeTo`, etc.) In addition, the pseudo link type `ComposedOf` is introduced to model the hierarchical decomposition of service objects. For directed links, only the "raw" link name (the "forward" direction) is used, i.e. **ITSMTrace** only knows about `DependsOn` but not about its reverse display name `RequiredBy`. Directed links are currently all assumed to point "down" in some underlying hierarchy of things. Using a type of directed link that points "up" might work, but is completely untested.

Constraints

To keep images reasonably sized and understandable, constraints allow to focus on relevant information. It is possible to include only specific object– and link types in the output, or to limit it to show just objects with a non-operational *IncidentState*.

It is also possible to trim the trace by only following directed links that point ''down'', and also stopping after following a non-directed link.

Clustering

To further enhance the understandability of the generated graph, objects are graphically clustered (boxed) with a service that depends on it. When the trace runs across a CI object that has a *direct* dependency link from a service that is *not* a descendant of the service that started the current cluster, the recursion is stopped under the assumption that this is another well-known building block.

Example:

If the MySQL Server as in the example of the ''Data Model'' section were a shared MySQL service with its own (unrelated) service entry, the trace would stop after the cluster for the MySQL service.

Web User Interface

ITSMTrace is integrated with the web UI of OTRS in the areas ''Service'' and ''CMDB''. By default, there is one link to show a trace of all objects of the given type. This trace tends to become unreadable even with a moderate number of objects.

The more valuable link is the one to show the trace for a single service or CI. By default, it comes with options configured to show a ''reasonable'' subset of the available information. Typical uses are to show the breakdown of a service into its constituent CIs or to show the services that depend on a given CI to be operational.

Command Line Interface

For full access to all capabilities of generating a trace, **ITSMTrace** also supplies a low-level command line interface in the `bin` directory. Output goes to the standard output.

KNOWN PROBLEMS
None yet.

POSSIBLE FUTURE ENHANCEMENTS
Without committing to anything on this list, the following features seem worthwhile to be followed up.

• Tuning of Graphviz Output
As it is now, links tend to be drawn very close to each other, which makes it difficult to follow them. Any graphviz specialist that would volunteer to help out on this? (Need to somehow increase spacing between ''ranks'' in graphviz).

• Access to more Information on the Web Page
The graphical output is currently a single PNG file. Provisions are already there to generate an associated imagemap that displays additional information when hovering over an object. Any web-programmer familiar with multi-part output or such stuff volunteering to jump in?

• Allow Names for Services/CIs in Command-Line Interface
Currently services or CIs can only be specified by their ID which is not really user–friendly.

• Allow Trace Options to be selected in the Web UI
The Web UI currently comes with predefined links that set a few options to reasonable values. This can be adapted through `SysConfig`, but is static from an agent's point of view. It would be useful to have an intermediate form page where the agent can individually select what to include in the output or not.

TERMS AND CONDITIONS
Copyright (C) 2011–2014 Thales Austria GmbH, http://www.thalesgroup.com/

This software comes with ABSOLUTELY NO WARRANTY and WITHOUT ANY SUPPORT.

For license information, see the enclosed file COPYING-ITSMTrace (GNU AFFERO GENERAL PUBLIC LICENSE, version 3). If you did not receive this file, see http://www.gnu.org/licenses/agpl−3.0.html.

AUTHOR

dietmar.berg@thalesgroup.com