Acme-layout – ACME Design Auto-Layout Tool

SYNOPSIS

```
acme-layout < input.acme > output.acme
```

DESCRIPTION

The ACME layout tool takes a valid ACME description as standard input and adds visual *properties* to the elements of the design which describe positioning and size information so that the design can be visualized. The new annotated design is then output to the standard output. The new design can then be read by a tool that requires the visual information. This includes environments like `Aesop` as well as ACME-based tools like the `acme-web` tool included in the ACME tools distribution. `Acme-web` can be used to generate HTML documentation based on the design, including diagrams that take advantage of the visual information:

```
acme-layout < somefile.acme | acme-webgen
```

The layout of components and connectors in Systems is done using graph layout algorithms from the Graphlet library (reference) which is compiled into the auto-layout tool. The current version of the tool uses a single algorithm from the library based on the (name) Mixed-Model algorithm (reference) . A future version of the acme-layout tool will support a choice of algorithms and way to specify options to those algorithms.

ACME INPUT RESTRICTIONS

Any valid ACME description can be used as input. No special properties are required. The current version of the tool ignores any visual properties already present in the input design. Future versions of this tool will take advantage of partial layout information, and allow additional layout constraints to be specified.

ACME OUTPUT PROPERTIES

The acme-layout adds the `vis-position` property to each System, Component, Connector, Port and Role in a design. The vis-position has the following ACME property type:

```
Property Type vis-positionT =
      Record [    x : float;
                  y : float;
                  width : float;
                  height :  float; ]
```

For example,

```
Component foo = {
      …
      Property vis-position =
            [ x = 50.0; y=50.0;width=100.0;height=50.0];
};
```

The fields `x` and `y` describe the coordinates of the *center* of a Component or Connector relative to the upper-left hand corner of the parent system in the usual pixel-based coordinate system.

For a Connector, the position refers to the "body" of the connector from which edges radiate to its roles. Port and Role positions are specified relative to the center of the parent Component or Connector, respectively.

The properties `width` and `height` describe the height and width of the bounding box of an element. For a connector, the properties refer to the size of the "body" of the connector. Note that the layout tool assumes that a component is rectangular when computing the positions of ports along its outside edge.

ENVIRONMENT

`acme-layout` does not use any environment variables, or external libraries.

SEE ALSO

The `acme-webgen` tool documentation.
The **ACME-Lib Programmer's Manual**.

Acme-webgen – ACME Design Web Documentation Tool

SYNOPSIS

acme-webgen *options* < *input.acme* > *output.acme*

DESCRIPTION

The acme-webgen tool takes an ACME description that has been annotated with properties describing the positions of elements in Systems (in the format output by `acme-layout`) and generates a documentation tree of design which can be viewed with a Web Browser.

Each element of the design is rendered as an HTML page that includes a graphical visualization of the element. The document appearance can be customized by defining a visual style in which to render a design. A visual style consists of HTML templates that describe the appearance of an element information page and descriptions of the graphical visualization to use for an element. .

OPTIONS

-name <name>

Rename the the top level system <name>. The top-level documentation directory will also have this name.

-style <style>

This option specifies the visual style in which to render the design. See below for details. It defaults to the value of ACME_STYLE.

-lib <lib-dir>

This option specifies the location of the visual style library. It defaults to the value of ACME_WEB_LIB.

-dest <destination-dir>

Destination directory for weblet in the current filesystem. Acme-webgen will create a directory tree rooted at a file created in this directory. If this option is not specified, the destination directory will default to the current directory, or to the value of the environment variable `ACME_DESTINATION`, if it is defined.

-map <imagemap-url>

Specifies the URL for a CGI imagemap script. Note that if a valid image-url is not defined, that images in the weblet will not be selectable in browsers that do not support client-side imagemaps. The WWW server that will be serving the documentation must have access to the file-system where the design will be stored. The URL defaults to the value of the environment variable ACME_IMAGEMAP.

-prefix <url-prefix>

This is the prefix that corresponds to the path the destination directory. This should correspond to the WebServer logical directory that corresponds to the directory in which the design is stored. It defaults to the value of the environment variable ACME_URL_PREFIX.

ACME INPUT PROPERTIES

The acme-webgen tool requires the following properties to be defined in an ACME description to be visualized:

All components, connectors, ports, roles must include properties describing their sizes and positions in the format described in the acme-layout documentation:

```
Property vis-position : vis-positionT =
      [ x = …; y = …; width = …; height = … ];
```

Each element may also define a visual style and class to be used:

```
Property vis-style : String = "…";
Property vis-class : String = "…";
```

RUN-TIME ENVIRONMENT

The following environment variables are read by acme-webgen (see the description of the program options above for a description of the variables):

```
ACME_DESTINATION
ACME_IMAGEMAP
ACME_STYLE
ACME_URL_PREFIX
ACME_WEB_LIB
```

The last variable ACME_WEB_LIB specifies the location of the visual style libraries used by the too. This directory contains three types of files:

*.style	Describes visualization styles, collections of visual classes which describe html page and visualization graphic for a visual class of element.
*.html	HTML template files that describe web pages to use for each element. One of these files is referenced in each visual class description in a *.style file.
*.vis	Descriptions of the the graphical visualization to use for an element. It describes the shapes and colors to use to visualize an element. One of these files is referenced in each visual class description in a *.style file.

VISUAL STYLES

An acme-webgen visual style library directory contains a series of files with the extensions ".style", each of which describes a different style of visualization. The following is an excerpt from the "generic" visual style included with the `acme-webgen` distribution:

```
# Generic visual Style definition
#
# Category   Visual Class     HTML Template File   Visual Desc. File
# ------------------------------------------------------------------
component   generic-comp      generic_system_comp_ws.html   comp.vis
connector   generic-conn      generic_conn_ws.html          conn.vis
system      generic-system    generic_system_conf_ws.html
role        generic-role      generic_role_ws.html          role.vis
port        generic-port      generic_port_ws.html          port.vis
```

There is an entry for each visual class in the style. Each visual class is associated with a Category: component, connector, system, port or role. A visual class can only be used to visualize an element in the same category.

Each visual class includes an HTML template file that describes a web page to use to document an element in the class. The HTML template file includes placeholders into which the tool will substitute different information associated with an element. The visual description file is a description of the shapes to use to render the element graphically.

The format of the file is as follows. Each file describes a particular style. Each line of the file (except commented lines beginning with #) describes a visual class and includes four space/tab delimited fields: category, visual class, template file and visual description file. Systems do not have visual description files.

FORMAT OF HTML TEMPLATE FILE

Each of the element types : System, Component, Connector, Port and Role are visualized as an HTML page in the weblet. An HTML file can be specified as the basis for the pages using the style file described above. The file is a regular HTML file that includes placeholders for the different types of information. Each placeholder is delimited by "%" signs. For example %vis% is a placeholder for the graphical visualization of the element.

All pages may include the following fields, `%PropertyList.typed%`, `%RepresentationList.typed%`, `%Name%`, `%vis%.` Acme-webgen will substitute the properties list, a list of representations, the name of the element, and the visualization for the element, respectively in those placeholders.

In addition, all Components, Connectors, Ports and Roles include a `%Type%` placeholder that will hold the name of the type of the constituent.

System pages can include the placeholders %Family% , %ComponentList.typed%, `%ConnectorList.typed%`, `%RoleList.typed%` and `%PortList.typed%`. These correspond to the family name and lists of the different sort of children. Similarly, component pages can include `%PortList.typed%` and connector pages, `%RoleList.typed%.`

Simple example HTML page for a component:

```
<html>

Component Name: %name% <P>
Component Type: %type%    <P>

Reps:<P>
%Representation.List.typed%<P>

Properties<P>

%PropertyList.typed%.

Ports:<P>
%PortList.typed%

</html>
```

See the example *.html files included in the distribution for more information.

FORMAT OF VISUAL DESCRIPTION FILE

To be written.