
CAIRIS User Manual

Release 2.3.8

Shamal Faily

Feb 01, 2023

Contents:

1	Quick Start	1
1.1	Live Demo	1
1.2	Video tutorials	1
1.3	Example models	1
1.4	Define your contexts of use	1
1.5	Save early and often	1
1.6	Supporting UX	2
1.7	Asset-driven security design	2
1.8	Threat-driven security design	2
1.9	Working with requirements	2
1.10	Thinking about architecture	2
1.11	Generating documentation	2
2	Installing CAIRIS	3
2.1	Installation via Vagrant	3
2.2	Installation via Docker	3
2.3	Installation and configuration via GitHub (automated)	5
2.4	Installation and configuration of server with account registration via GitHub (automated)	5
2.5	Installation and configuration via GitHub (manual)	6
3	Starting CAIRIS	9
3.1	Starting the CAIRIS server	9
3.2	Starting the web application	9
4	CAIRIS databases	13
4.1	Default database	13
4.2	Using other databases	13
4.3	Providing database access to other users	14
5	Reserved characters in object names	15
6	Importing and Exporting models	17
6.1	Importing models	17
6.2	Exporting models	18
7	Sample models	21
7.1	Exemplars	21
7.2	Threat and Vulnerability Directories	22
8	Creating a new project	23
9	Environments	25

9.1	Adding a new environment	25
10	Assets	27
10.1	Adding, updating, and deleting an asset	28
10.2	Asset modelling	29
11	Roles	35
11.1	Adding, updating, and deleting a role	35
11.2	Responsibility modelling	36
12	Personas	39
12.1	Adding, updating, or deleting a persona	39
12.2	Assured personas with persona characteristics	40
12.3	Automating persona characteristic creation	42
13	Tasks	51
13.1	Adding, updating, or deleting a task	51
13.2	Task traceability	53
13.3	Visualising tasks	53
14	Misusability Cases	55
14.1	Creating concept references	55
14.2	Creating the skeleton scenario	55
14.3	Creating task characteristics	55
14.4	View misusability case models	55
15	Domain Properties	57
15.1	Adding, updating, and deleting a domain property	57
16	Goals, Requirements, and Obstacles	59
16.1	Adding, updating, and deleting a goal	59
16.2	Goal Modelling	61
16.3	Adding, updating, and deleting an obstacle	62
16.4	Obstacle Modelling	64
16.5	Adding, updating, and deleting requirements	65
16.6	Visualising Requirements Quality using Chernoff Faces	65
16.7	Attack tree modelling with obstacles	67
17	Use Cases	69
17.1	Adding, updating, or deleting a use cases	69
17.2	Add exceptions to use case steps	73
18	User goals and user goal models	75
18.1	Adding, updating, and deleting user goals	75
18.2	Adding, updating, and deleting user goal contributions	76
18.3	Task contributions	77
18.4	Adding User goal elements to persona characteristics	77
18.5	Adding GRL elements to use cases (jUCMNav export only)	79
18.6	Viewing a user goal model	81
18.7	Working with workbooks	81
18.8	Generating a jUCMNav compatible GRL model	82
19	Dependencies	85
19.1	Adding, updating, and deleting a dependency	85
19.2	Viewing dependencies	85
19.3	Introducing Personal data into CAIRIS using dependencies	86
20	Security Patterns	87
20.1	Create a template asset	87
20.2	Create a security pattern	88

20.3	Situate a security pattern	90
21	Vulnerabilities	91
21.1	Create a vulnerability	91
21.2	Introducing template threats and vulnerabilities	92
22	Attackers	95
22.1	Adding, updating, and deleting an attacker	95
23	Threats	97
23.1	Adding, updating, and deleting a threat	97
24	Threat Modelling	99
24.1	Data flows and Data Flow Diagrams	99
24.2	Attack trees	103
25	Using CAIRIS as tool-support for STPA	107
25.1	Overview	107
25.2	Step 1: Define purpose of the analysis	107
25.3	Step 2: Model the control structure	108
25.4	Step 3: Identify unsafe control actions	109
25.5	Step 4: Identify loss scenarios	109
25.6	Supporting other STPA outputs	109
26	Modelling access control needs and policies	111
26.1	Overview	111
26.2	Modelling access needs	111
26.3	Modelling access control policies with policy statements	112
26.4	Access control model validation checks	113
27	Risks	115
27.1	Adding, updating, and deleting a risk	115
27.2	Risk Analysis model	116
28	Locations	121
28.1	Adding, updating, and deleting a locations object	121
28.2	Viewing location models	122
29	Risk Responses	125
29.1	Adding, updating, and deleting a response	125
29.2	Generating goals	126
30	Countermeasures	127
30.1	Adding, updating, and deleting a countermeasure	127
30.2	Generating countermeasure assets and security patterns	130
30.3	Associating countermeasures with pre-existing patterns	130
30.4	Weakening the effectiveness of countermeasures	130
30.5	Mitigating weakening effects	130
31	Traceability	133
31.1	Allowable manual traceability links	133
31.2	Editing manual traceability links	133
31.3	Visualising manual traceability links	134
32	Architectural Patterns	137
32.1	Editing Architectural Patterns	137
32.2	Viewing Architectural Patterns	139
32.3	Situating a pattern	139

33	Model Validation	145
33.1	General validity checks	145
33.2	Security design checks	146
33.3	Privacy design checks	146
33.4	Access control checks	146
34	Configurable Types and Values	149
34.1	Asset Values	149
34.2	Asset Types	149
34.3	Vulnerability and Threat Types	149
34.4	Other Types	149
35	Searching model objects	151
36	Tags	153
37	Generating Documentation	159
37.1	Problems with wide models	159
37.2	Customising model files	160
38	CAIRIS server maintenance	161
38.1	Account management	161
38.2	Importing and exporting models	161
38.3	Backing up and restoring servers	162
39	Using the CAIRIS API	163
39.1	API documentation	163
39.2	Authenticating with the CAIRIS server	163
39.3	The cairis_test database	164
40	Extending CAIRIS	167
40.1	1. Define the database tables	167
40.2	2. Define the database procedures	167
40.3	3. Update the Python database proxy	167
40.4	4. Write your model object test case	167
40.5	5. Update the CAIRIS DTDs	168
40.6	6. Update the model import / export code	168
40.7	7. Implement the server end-points	168
40.8	8. Write your API test case	168
40.9	9. Update the UI	168
40.10	10. Update the documentation generation process	169
41	Troubleshooting	171
41.1	Log files	171
41.2	Raising issues	171
42	Indices and tables	173

1.1 Live Demo

A live demo of CAIRIS is available to use on <https://demo.cairis.org>.

The demo has a test account (user: *test@test.com*, password: *test*) with two example databases you can explore: [NeuroGrid](#), [ACME Water](#). You are also free to create your account to explore CAIRIS' capabilities on your own.

The live demo is rebuilt every night based on the latest updates to CAIRIS, so please feel free to add, update, or remove elements in the example models. The test account is dropped and re-created each night with the sample models. Other accounts created on the server are dropped on Sunday morning each week.

1.2 Video tutorials

The [CAIRIS YouTube channel](#) has several short video primers. These include an overview of the UI, and guidance on using CAIRIS for different design activities.

1.3 Example models

1.4 Define your contexts of use

How you use CAIRIS depends on how you approach the early stages of your design. You will, however, need to work with *environments* to represent your contexts of use. Each model comes with a *Default* environment, but you may wish to add more later as you learn more about different contexts.

1.5 Save early and often

You should *save* your working model early and often. Saving a model in CAIRIS entails exporting it. CAIRIS models are XML, so easy to edit using other tools and easy to version control.

1.6 Supporting UX

CAIRIS supports the creation and management of personas to represent archetypical users, and *tasks* to describe how these interact with the system being designed. You need to define roles that the personas fulfil before creating personas, and personas before creating tasks. As your design evolves *task models* and *risk analysis models* will summarise the impact that security and usability are having on each other.

1.7 Asset-driven security design

Once you've specified at least one environments, you can start modelling *assets* : the things that are important to you. You should model relationships between them to help you make sense of your growing design, and identify new assets you need to protect. As asset models gives you ideas about possible system weaknesses, record these as *vulnerabilities*. As you think of new threats, note who you think the *attacker* might be, and what threats they might carry out. Armed with these insights, you can then create *risks* that bring everything together. Based on these risks, you can decide how to *respond* and add *countermeasures* to mitigate them.

1.8 Threat-driven security design

You don't have to start your design by thinking about assets. CAIRIS encourages the early creation of *threat models*, which can be useful if you're still trying to make sense of what the system is and how attackers might exploit it. This can help you better understand what your assets are, and even help you understand what the usability implications of certain threats might be.

1.9 Working with requirements

The earlier you start finding *requirements*, the easier it will be to spot other issues in your design. CAIRIS lets you model requirements as goals, requirements, and use cases.

1.10 Thinking about architecture

Requirements aren't always easy to find, and sometimes thinking about possible architectures can help you work backwards. You can use *architectural patterns* as building blocks and introduce these into environments to see risks they might be exposed to, or how they might impact personas and tasks. You can also use *security patterns* to see what their consequences of different pieces of *best practice* might have on your design.

1.11 Generating documentation

Your stakeholders may not want to work directly with CAIRIS, so you can *generate documentation* to share your design documentation with others.

Installing CAIRIS

2.1 Installation via Vagrant

If you have [Vagrant](#) and [VirtualBox](#) installed, you can build your CAIRIS VM in minutes. To start, you need to clone the cairis server repository:

```
git clone https://github.com/cairis-platform/cairis
```

Once in the root directory of the repository type:

```
vagrant up
```

This will create and start a CAIRIS virtual machine in VirtualBox and, once complete, this is accessible via your web browser at <http://localhost:7071>. The default username and password is *test@test.com* and *test*, but you can change this by editing *vagrant_conf.yaml*.

To shutdown the virtual machine:

```
vagrant halt
```

To restart the virtual machine:

```
vagrant up
```

If you need to login to the virtual machine, i.e. to check the log files, use the *vagrant* account (password: *vagrant*).

Note: The Vagrantfile is a simplified version of Ben Coleman's [Cairis_vagrant](#) repository.

2.2 Installation via Docker

If you have Docker installed on your laptop or an available machine, you can download the CAIRIS container from [Docker hub](#). Like the live demo, this is built from the latest version of CAIRIS in GitHub, and uses [mod_wsgi-express](#) to deliver the CAIRIS web services.

There are two options for running the container, a full install of everything or a smaller install which doesn't provide pdf export functionality:

For the full install (with pdf export functionality) download and run the container, the documentation container, and its linked mysql container:

```
sudo docker run --name cairis-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d_
↳mysql:latest --thread_stack=256K --max_sp_recursion_depth=255 --log_bin_trust_
↳function_creators=1
sudo docker run --name cairis-docs -d -v cairisDocumentation:/tmpDocker -v_
↳cairisImage:/images -t shamalfaily/cairis-docs
sudo docker run --name CAIRIS -d --link cairis-mysql:mysql --link cairis-docs:docs_
↳-P -p 80:8000 --net=bridge -v cairisDocumentation:/tmpDocker -v cairisImage:/
↳images shamalfaily/cairis
```

For the smaller install (without pdf export functionality) download and run the container, and its linked mysql container:

```
sudo docker run --name cairis-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d_
↳mysql:latest --thread_stack=256K --max_sp_recursion_depth=255 --log_bin_trust_
↳function_creators=1
sudo docker run --name CAIRIS --link cairis-mysql:mysql -d -P -p 80:8000 --
↳net=bridge shamalfaily/cairis
```

If you run the above commands on macOS (and possibly other non-Linux platformns), you might get the error *links are only supported for user-defined networks*. If so, you should instead run the below commands to download and run your containers:

```
NET=cairisnet
docker network create -d bridge $NET
docker run --name cairis-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:latest_
↳--thread_stack=256K --max_sp_recursion_depth=255 --log_bin_trust_function_
↳creators=1
docker network connect $NET cairis-mysql
docker run --name CAIRIS -d -P -p 80:8000 --net=$NET shamalfaily/cairis
```

If you want to use the containers to support account self-registration and revocation then you can set MAIL_SERVER, MAIL_PORT, MAIL_USER, and MAIL_PASSWD environment variables to correspond with the SSL outgoing mail server, mail server port, mail account username, and password for the mail account, i.e.

```
docker run --name CAIRIS --env MAIL_SERVER=mymailserver.com --env MAIL_PORT=465 --
↳env MAIL_USER=admin@mymailserver.com --env MAIL_PASSWD=mypassword -d -P -p_
↳80:8000 --net=$NET shamalfaily/cairis
```

The *docker run* commands will create and start-up CAIRIS. If you haven't setup account self-registration then you will need to create an account before you can use it. To do this, run the below command - replacing *test@test.com* and *test* with your desired username and password.

```
docker exec -t `docker ps | grep shamalfaily/cairis | head -1 | cut -d ' ' -f 1` /
↳addAccount.sh test@test.com test TestUser
```

If you are using PowerShell on Windows to run the above command then this might fail because *grep* is not installed. To work around this, you need to use *docker ps* to get the Container ID and run the below modified command:

```
docker exec -t CONTAINER_ID /addAccount.sh test@test.com test TestUser
```

Once the containers have been installed then, in the future, you should use *docker start* rather than *docker run* to start up the already downloaded containers.

```
sudo docker start cairis-mysql
sudo docker start CAIRIS
```

The containers can be stopped using *docker stop*, i.e.

```
sudo docker stop CAIRIS
sudo docker stop cairis-mysql
```

To update your docker containers, stop the docker containers and run the below commands to remove any old containers and volume files. Following that, you can re-run the above *docker run* commands to install and run the container. Don't forget to re-add your user account!

```
sudo docker rm $(sudo docker ps -aq)
sudo docker rmi --force $(sudo docker images -q)
sudo docker volume rm $(docker volume ls)
```

2.3 Installation and configuration via GitHub (automated)

If you have a clean Ubuntu VM, you can quickly install and configure CAIRIS and its dependencies with the command below, replacing my-secret-pw with your desired root password for MySQL.

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y &&
↪sudo apt install curl -y && sudo apt install net-tools -y && curl -s https://
↪cairis.org/quickInstall.sh | bash -s my-secret-pw
```

In addition to configuring and installing CAIRIS, the script creates an initial user account (username: test@test.com, password: test), starts the Flask development server as a service, and restarts the VM. You can use *journalctl* to check the CAIRIS log file.

```
journalctl -u cairis.service -f
```

This script also adds an alias so, in future, you can update CAIRIS by running the below command:

```
update_cairis
```

2.4 Installation and configuration of server with account registration via GitHub (automated)

If you have a clean Ubuntu VM, want to quickly install CAIRIS for multiple users, but don't want to use the defaults associated with the quickInstall.sh script, then you can run the more bespoke serverInstall.sh script as below, replacing (i) my-secret-pw with your desired MySQL root password, (ii) mymailserver.com with the name of your private (with SSL) outgoing mail server, (iii) 465 with this mail server's port, (iv) admin@mymailserver.com with your mail server username, and (v) mypassword with this account's password.

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y &&
↪sudo apt install curl -y && sudo apt install net-tools -y && curl -s https://
↪cairis.org/serverInstall.sh | bash -s my-secret-pw mymailserver.com 465
↪admin@mymailserver.com mypassword
```

When working with very large models, you may get memory errors when viewing goal models or carrying out model validation checks. If you do, you could consider increasing the thread_stack size in */etc/mysql/conf.d/mysql.cnf*. For example, increasing the size to 1024K made it possible to valid even really big system-of-system models, but you can increase or decrease this size based on your server's performance and the number of users you expect the server to support.

If you follow these instructions then, once you've restarted your server, CAIRIS should be accessible via <http://SERVER:8000>, where SERVER is the name or IP address of your machine. If you wish to route your http traffic accordingly (e.g. via DNS) then the CAIRIS service supports access via https too. This is the approach currently taken by the CAIRIS live demo on <https://demo.cairis.org>.

Although no `update_cairis` alias is created, we provide a `rebuildServer.sh` script which, if run from cron each night, will rebuild and reconfigure CAIRIS while still retaining the user accounts and their default databases created on the server. This script takes the same command line arguments as the `serverInstall.sh` script, with the addition of additional arguments for the name of the account running CAIRIS, and the accounts home directly. For example, if the account running CAIRIS is `sfaily` and the home directory in `/home/sfaily` then, to rebuild the server at 0200 each morning you should run `sudo crontab -e` and add the following line to your crontab:

```
0 2 * * * /home/sfaily/rebuildServer.sh my-secret-pw mymailserver.com 465_
↪admin@mymailserver.com mypassword sfaily /home/sfaily > /home/sfaily/rebuild.log_
↪2>&1
```

This `rebuild.log` file should be useful for troubleshooting any problems with the rebuild.

Once the server is running, users can register for accounts using the Register link on the login page. The account name should be a valid email address. When an account is created, an email is sent to the user and the user is logged in. If the Reset link is clicked and the account name is provided, CAIRIS will email instructions for resetting the password to the user.

2.5 Installation and configuration via GitHub (manual)

If you're happy to use the command line, you may like to install CAIRIS from the latest source code in GitHub. CAIRIS can be installed on any platform that its open-source dependencies are available for. The most tested platform is **Ubuntu**. Assuming you are using Ubuntu, just follow the steps below:

Begin by installing the required applications and dependencies:

```
sudo apt-get install python3-dev build-essential mysql-server mysql-client_
↪graphviz docbook dblatex python3-pip python3-mysqldb python3-numpy git_
↪libmysqlclient-dev --no-install-recommends texlive-latex-extra docbook-utils_
↪inkscape libxml2-dev libxslt1-dev poppler-utils python3-setuptools pandoc
```

If you are installing Ubuntu 18.04 LTS or later, or have not been prompted to set a root database password, you will need to set this manually. This entails starting `mysqld` with the `--skip-grant-tables` option, logging into `mysql` as `root`, and setting the root password by hand. You can find instructions on how to do that [here](#).

In addition to the above, you also need to update my MySQL server system variables. You can do this by adding or updating the below values to your `mysqld.cnf` file. In Ubuntu 19.04, you can find this in `/etc/mysql/mysql.conf.d`, but the file location might differ depending on your OS and MySQL version:

```
thread_stack = 256K
max_sp_recursion_depth = 255
log_bin_trust_function_creators = 1
```

Clone the latest version of the CAIRIS github repository, and use `pip` to install the dependencies in the root directory, i.e.

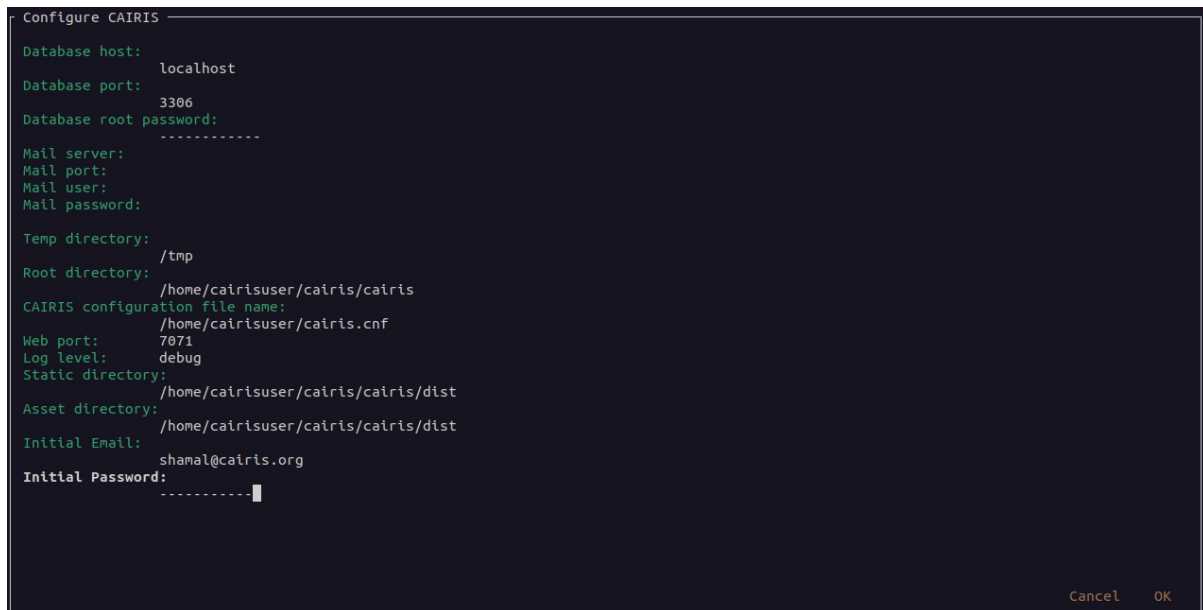
```
git clone https://github.com/cairis-platform/cairis
cd cairis
sudo pip3 install -r requirements.txt
```

Run the CAIRIS quick setup initialisation script (which can be found in `cairis/`). When you run this script, you should get the below form.

```
./quick_setup.py
```

You can accept many of these defaults, except for the database root password, an initial username and password which need to be supplied. Please note that the username `root` is reserved, so you should not use this.

Details for the mail server only need to be set if you intend to provide self-service registration and reset of accounts. This is typically only necessary if you plan to install CAIRIS to a server for multiple users. When these settings are set, the mail server and port should be for out-going SSL traffic.



If you want more diagnostic information logged, you find it useful to change the Log Level from *warning* to *debug*.

The static and directory and asset directory will point to the location of UI code, but these directories will not be created during this step. If you don't plan to customise your web server setup, you should retain these default values.

When you select *Ok*, the script will create a new CAIRIS database, and accompanying CAIRIS configuration file; this file will ensure that CAIRIS knows what database it needs to refer to when you start up the tool and setup the necessary environment variables.

Logout of your current account or, alternatively, reload your `.bashrc` file i.e.

```
source .bashrc
```

The final step entails installing the UI code by running the below script in `cairis/cairis/bin`

```
sudo -E ./installUI.sh
```

The CAIRIS UI code is managed in the [cairis-ui github repository](#). Running this script will setup `node` and `yarn`, download the github repo, create a production version of the latest UI code and deploy to `cairis/cairis/dist`. The `-E` flag is required, as the `CAIRIS_SRC` environment variable needs to be visible to root.

You should now start up your CAIRIS server. If you plan to develop with CAIRIS, you should skip this step as you'll find it more useful to manually start the Flask development server. For everyone else, create the following `cairis.service` file, substituting `cairisuser` for the name of your account. Using `sudo` or `root`, copy this file to `/etc/systemd/system`.

```

[Unit]
Description=cairisd

[Service]
User=cairisuser
WorkingDirectory=/home/cairisuser/cairis
Environment="FLASK_APP=/home/cairisuser/cairis/cairis/daemon:create_app"
Environment="FLASK_ENV=development"
Environment="CAIRIS_CFG=/home/cairisuser/cairis.cnf"
Environment="PYTHONPATH=${PYTHONPATH}:/home/cairisuser/cairis"
ExecStart=flask run --host 0.0.0.0 --port 7071
Restart=on-failure

```

(continues on next page)

(continued from previous page)

```
[Install]
WantedBy=multi-user.target
```

You can now launch `cairisd` as a system service:

```
sudo systemctl enable --now /etc/systemd/system/cairis.service
```

[Optional] Multiple users using CAIRIS

`cairisd` relies on the Flask development server, which is fine for a single user, or development and troubleshooting. However, if multiple users will use the same CAIRIS service at once, or you want to run CAIRIS in a production environment then it may be sensible to use `mod_wsgi-express` instead. To do this, you will need to install the requisite Apache2 packages.

```
sudo apt-get install apache2 apache2-dev
```

You will then need to use `pip` to install the requisite dependencies.

```
sudo pip3 install -r wsgi_requirements.txt
```

You should then use `mod_wsgi-express` to run `cairis.wsgi` (also in `cairis/cairis/bin`):

```
mod_wsgi-express start-server cairis.wsgi
```

Don't forget to modify `cairis.service` accordingly!

[Optional] Additional steps for developers

If you plan to customise CAIRIS, development extensions or fixes, you should install the requisite packages for running the tests in `cairis/cairis/test`.

```
sudo pip3 install -r test_requirements.txt
```

To start the CAIRIS development server, set the `FLASK_APP` environment variable to `cairis/cairis/daemon:create_app`, the `FLASK_ENV` environment variable to `development`, then run:

```
flask run --port 7071
```

All logged output is sent to the console where you started the development server, which is useful when it come to diagnosing any problems. Also, if you plan to use `pytest` to debug any CAIRIS server code (i.e. by adding `import pytest` and `pytest.set_trace()` before any code you want to debug), the debug prompt will appear in the console.

Starting CAIRIS

3.1 Starting the CAIRIS server

If you are using Docker then the command used to install the container also starts the CAIRIS server on port 80.

If you are the only person that plans to use CAIRIS, using the Flask development server should be sufficient. Once the `FLASK_APP` and `FLASK_ENV` environment variables have been set, you can run:

```
flask run --port 7071
```

If you plan to use `mod_wsgi-express` then you need to use `cairis.wsgi` (also in `cairis/cairis/bin`):

```
mod_wsgi-express start-server cairis.wsgi
```

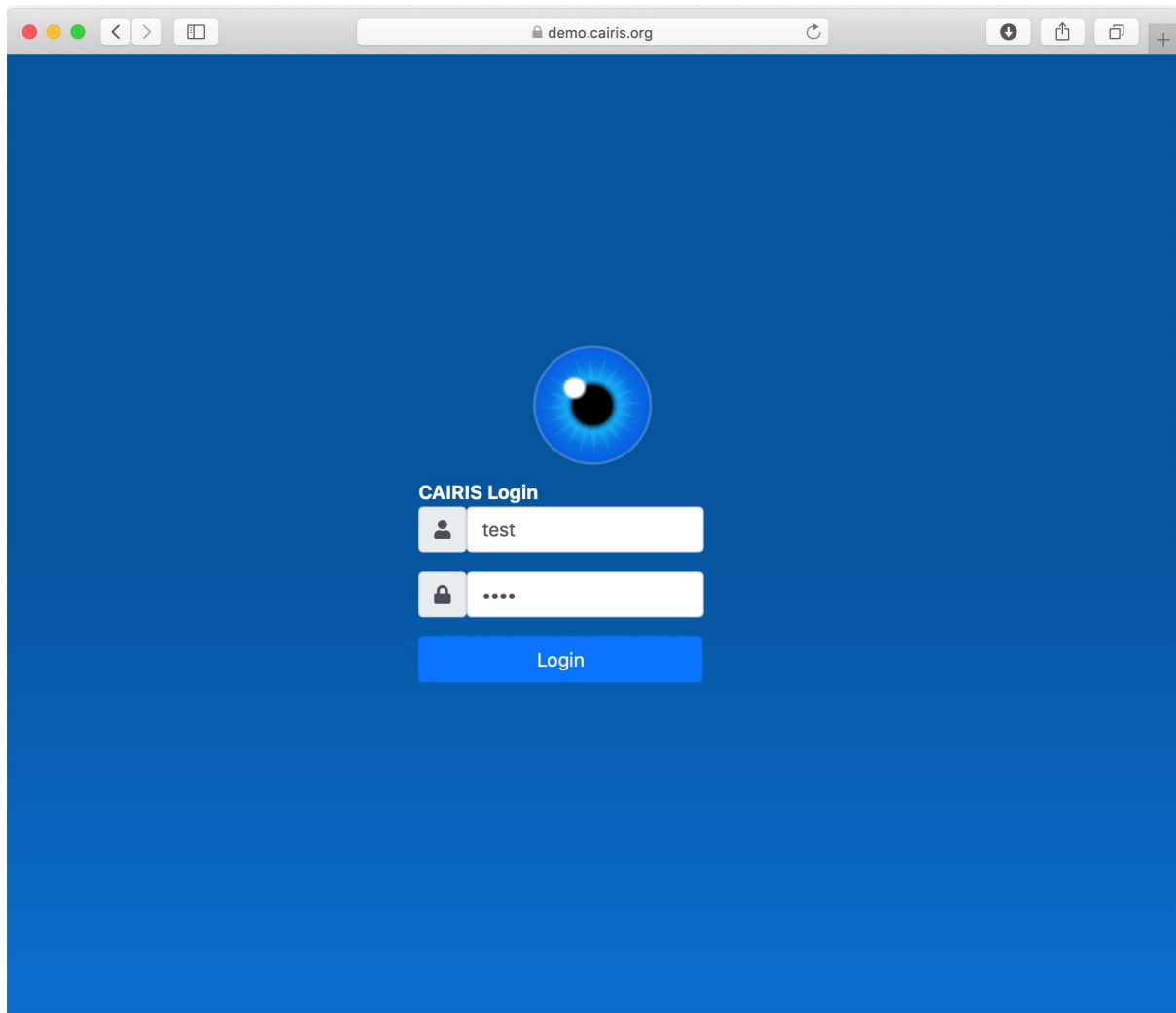
3.2 Starting the web application

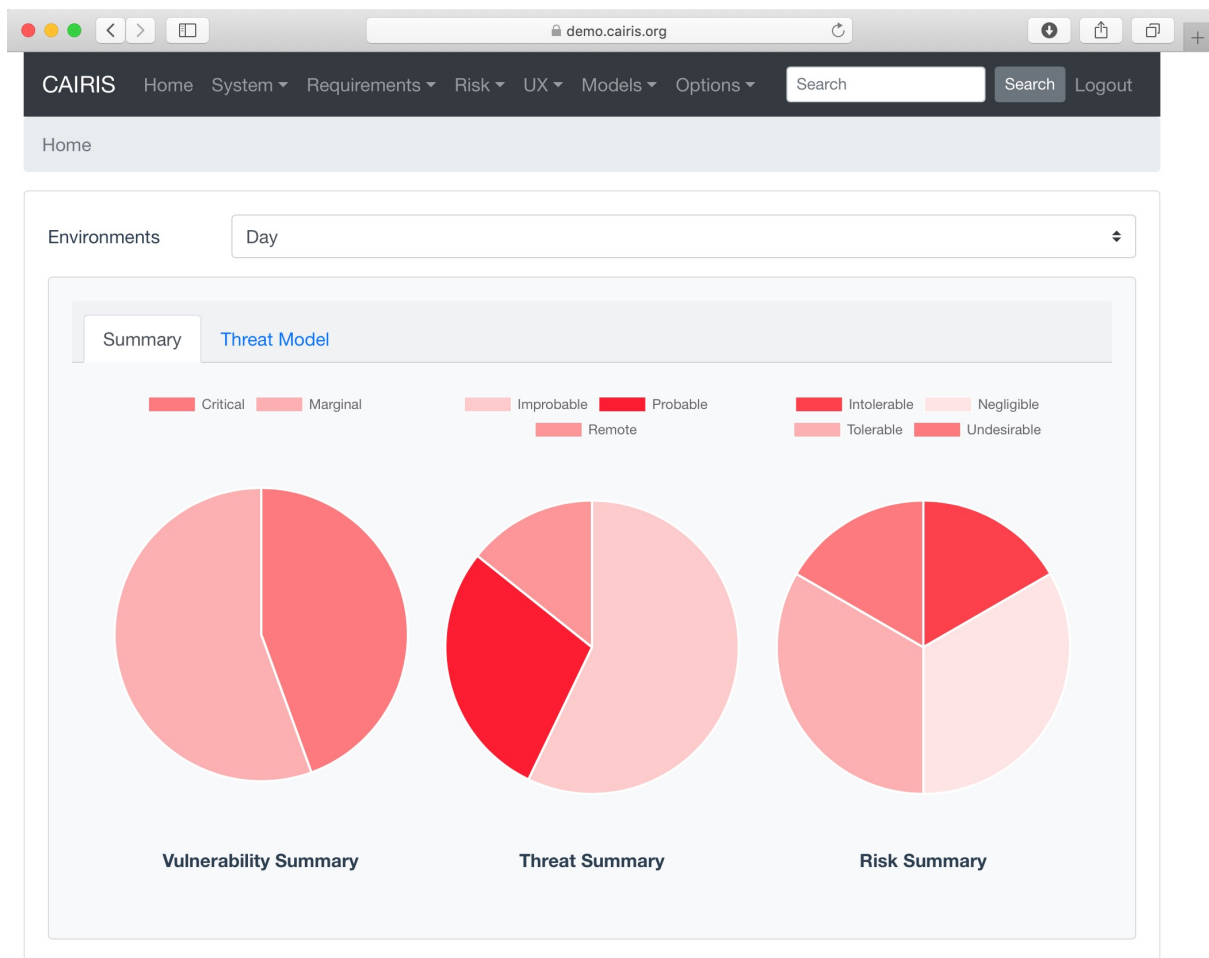
You can use CAIRIS on any modern web browser except Microsoft Internet Explorer (although you can use Microsoft Edge).

In your browser, visit the site hosting the CAIRIS server, and authenticate using credentials you have, or setup if you ran the `quick_setup.py` script. If you are not using the live demo, or have not mapped `mod_wsgi-express` to port 80, you will need to also specify the port the CAIRIS server is listening on. If you don't specify otherwise, `cairisd` will listen on port 7071, and `mod_wsgi-express` will listen on port 8000. For example, if you are using `cairisd` on `germaneriposte.org` then you should connect to <http://germaneriposte.org:7071>

Once you login in you should see the home page, which provides a summary of threats, vulnerabilities and risks, and the threat model for different environments.

Once you have finished working with CAIRIS, click on the Logout button.





CAIRIS databases

4.1 Default database

Each CAIRIS account comes with an *default* database. If you or your team are using CAIRIS to work on a single project at any given time then you shouldn't need to worry about additional databases if you are using the same account.

4.2 Using other databases

There might be times when it might be helpful to setup multiple databases. For example, the live demo on <https://demo.cairis.org> has two exemplar databases that people can interact with to see different examples of CAIRIS projects.

To create a new database, select the System/Databases menu, click on the Add button in the databases table, and enter the name for a new database. The name must not contain any spaces or reserved characters. After a few moments, a new database will be created and your CAIRIS application will point to this database. Any databases you create will be visible only to your account.

Note: Based on the configuration of MySQL, you may find that - on creating a new database - you no longer see the default database in your database list. If this is the case, you should logout of CAIRIS and log back in to return to the default database.

To open another database, select the System/Databases menu, and click on the table row corresponding with the name an existing database. After a few moments, your CAIRIS application will point to the chosen database.

You can delete a database by selecting the System/Databases menu, and clicking on the Delete button next to the database you want to remove. You cannot delete the database you currently have open.

To empty the contents of a currently open database, select the Systems/Databases menu and click on the Clear Current button.

4.3 Providing database access to other users

If you have created a non *default* database, you can grant or revoke access to other users by clicking on the Permissions button. Adding other users to the permissions list grants access, and removing them revokes access.

Note: Based on the configuration of MySQL, you may find the list of users with permission to access the database may not always show correctly once a permission has been added. If this is the case, and the user granted access to a database is unable to access it, you can manually grant or revoke permissions on the server using the `dbctl.py` script in `cairis/cairis/bin`, e.g. `./dbctl.py --database MySharedDB --user shamal.faily@gmail.com --privilege grant`.

Reserved characters in object names

When creating most new objects in CAIRIS, you need to provide a name.

The following characters are considered reserved and should be avoided when defining any object name: < > ‘ ’ “ ” \ : % _ * / ? # £ \$ & . You should also avoid using any non-ASCII characters.

The CAIRIS UI should warn you if you are about to create or update an object with a reserved character. If your object includes these characters, it may be possible for you to add the object, but you may get problems update or deleting them, or exporting model files containing the objects

When using the [Persona Helper](#), you may be diligently ensuring that spurious characters like ampersands don't creep into your factoid names. However, you may not notice that the web page you work with may contain reserved characters like ampersands and, once you create a factoid from the page, an external document will be created in CAIRIS containing the reserved character/s. This doesn't cause any problems while working with your model or even exporting it, but you will likely get errors about your model not being 'well-formed' when trying to import it back into CAIRIS.

There are no easy ways of getting around this problem in the Persona Helper extension, but there are two easy ways within CAIRIS itself to avoid or work-around this problem.

1. Go to the UX/External Documents menu and, if you see any external documents with reserved characters in their names, simply remove them from CAIRIS.
2. If you forget to do this and discover an error when importing the model file, you can easily remove the offending characters from the model file itself. If you have exported your model as 'Model (XML file)' then, you can use a tool like xmllint or one of the several free online XML validators available, such as FreeFormatter to check your model. This will flag invalid XML that you should remove or reword. If you have exported your model as 'Model' then you need to (i) unzip the model file, (ii) repeat the above step for the model.xml file, (iii) re-zip the model file as a .cairis file.

Importing and Exporting models

6.1 Importing models

You can import models by selecting the System / Import Model menu, selecting the model type to import, and the model file itself.

The screenshot shows the CAIRIS web application's 'Import' interface. At the top is a dark navigation bar with the CAIRIS logo and a menu: Home, System, Requirements, Risk, UX, Models, Options. To the right of the menu are a search bar with a 'Search' button and a 'User' dropdown. Below the navigation bar is a light gray breadcrumb bar showing 'Home / Import'. The main content area is a light gray box containing two sections. The first section, labeled 'Model', has a dropdown menu currently showing 'Model package (.cairis)'. The second section, labeled 'File', contains a 'Choose file' button and the text 'No file chosen'. At the bottom of the main content area are two buttons: 'Import' (blue) and 'Cancel' (gray).

You will usually want to stick with the *Model package (.cairis)* option to import .cairis files. .cairis files are zip archives with a model file, any supplemental locations and architectural pattern models, and all the image files associated with the model.

You can, alternatively, select the *Model file (.xml)* option, which imports a standard CAIRIS XML model file (as defined by the DTD in https://cairis.org/dtd/cairis_model.dtd). If you select this option, you can choose to overwrite an existing model (the default option) or you can incrementally import the contents of a model file into a pre-existing model.

You can also import other types of model into your current working project.

Model type	DTD (in https://cairis.org/dtd)	Model elements
Project data	cairis.dtd	Project background, goal, scope, rich picture, naming conventions, contributors, revisions
Requirements	goals.dtd	domain properties, goals, obstacles, requirements, use cases, and countermeasures
Risk analysis	riskanalysis.dtd	roles, assets, vulnerabilities, attackers, threats, risks, responses, asset associations
Usability	usability.dtd	personas, external documents, document references, concept references, persona characteristics, task characteristics, tasks
Misusability	misusability.dtd	concept references, task characteristics
Associations	associations.dtd	manual associations, goal associations, dependencies
Threat and Vulnerability Types	tvtypes.dtd	vulnerability types, threat types
Domain Values	domainvalues.dtd	threat values, risk values, countermeasure values, security values, likelihood values, motivation values, capability values
Threat and Vulnerability Directory	directory.dtd	vulnerability directory entries, threat directory entries
Security Pattern	securitypattern.dtd	security patterns
Architectural Pattern	architectural_pattern.dtd	architectural patterns
Attack Pattern	attack_pattern.dtd	attack patterns
Synopsis	synopsis.dtd	characteristic synopses, reference synopses, step synopses, reference contributions, usecase contributions
Assets	template_assets.dtd	template assets
Processes	processes.dtd	CSP process elements (used by desktop application only)
Locations	locations.dtd	locations
Dataflows	dataflow.dtd	dataflows and trust boundaries
Stories	stories.dtd	User stories
Attack Tree (Dot)	N/A	Graphviz (Dot) representation of an attack tree
diagrams.net (Data Flow Diagram)	N/A	diagrams.net drawn DFD
diagrams.net (Asset Model)	N/A	diagrams.net drawn asset model
User goals (Workbook)	N/A	CAIRIS generated Excel workbook with user goals and contributions

6.2 Exporting models

CAIRIS Home System Requirements Risk UX Models Options Search User

Home / Export

Model

☒ Model package (.cairis)
 ☐ Model file (.xml)
 ☐ GRL
 ☐ Architectural Pattern
 ☐ Security Patterns
 ☐ Persona characteristics (Workbook)
 ☐ User goals (Workbook)

File name

model

Export Cancel

To export a model, select the System / Export Model option. Exporting the current model renders the current CAIRIS database you are working with as a CAIRIS XML model (conforming to `cairis_model.dtd`). You can also

export a selected architectural pattern, the security patterns currently loaded into CAIRIS, or a GRL model for a selected environment and task; this GRL can be imported into jUCMNav.

7.1 Exemplars

CAIRIS comes with three complete system specifications. These illustrate how CAIRIS can be used, and – in some cases – provide templates to inspire your own use of the platform. These specifications are .cairis files can be found in the `cairis/examples/exemplars` directory, but their component model files and images can be found in sub-directories within that directory.

7.1.1 NeuroGrid

NeuroGrid is a data grid for neuroscience research. The sensitive of clinical data processed by NeuroGrid and its distributed nature drives the need to find secure and effective ways of accessing and managing it. This exemplar is restricted to the upload and download of data to and from NeuroGrid. This exemplar also comes with a physical locations file (Computing Laboratory) and an architectural pattern (WebDAV).

7.1.2 ACME Water

ACME Water is a fictional water company concerned with the delivery of wastewater and cleanwater services in a specific geographic region of the UK. This exemplar specifies a secure operating environment for SCADA, telemetry, and control systems associated with assets owned and operated by ACME. This exemplar also comes with a physical locations file (Poole Waste Water Treatment Works).

7.1.3 webinos

The [webinos](#) platform is a software runtime environment that allows the discovery of devices and services based on technical and contextual information. It exposes a set of APIs that provide access to cross-user, cross-service, and cross-device functionality. Unlike the other examples, the constituent CAIRIS models were generated from a variety of formats including spreadsheets, text files, and multiple smaller CAIRIS model files. You can find this design data and the scripts used to generate the model at [webinos-design-data GitHub repository](#).

7.2 Threat and Vulnerability Directories

These are libraries of importable threats and vulnerabilities, and can be found in the `cairis/examples/directories` directory.

7.2.1 CWE/CAPEC

`cwecapec_directory.xml` contains a selection of threats and vulnerabilities from CWE and CAPEC. To import this, it is first necessary to import `cairis/examples/threat_vulnerability_types/cwecapec_tv_types.xml`.

7.2.2 ICS Protection Profile

`ics_directory.xml` contains a selection of threats and vulnerabilities from the System Protection Profile - Industrial Control Systems issued by NIST. To import this, it is first necessary to import `cairis/examples/threat_vulnerability_types/ics_tv_types.xml`.

7.2.3 OWASP

`owasp_directory.xml` contains a selection of threats and vulnerabilities drawn from the OWASP body of knowledge. To import this, it is first necessary to import `cairis/examples/threat_vulnerability_types/owasp_tv_types.xml`.

Creating a new project

The first stage of any design process involves establishing the scope of subsequent analysis. CAIRIS supports this exercise by using the Properties form.

- Select the System/Properties menu to open the Project Settings notebook. By default, the notebook will open in the Background page. Enter the project name and background in this page.
- Click on the Goals tab and enter the high-level goals of that the system being specified needs to satisfy.
- Click on the Scope tab and enter the scope of the system being specified.
- If a rich picture or context diagram has been agreed, click on the Rich Picture tab, and click on the image (or avatar if no rich picture has been defined) to import. Permitted image types are jpg, png, gif, and bmp.
- Names or terms that the readership of the specification may be unfamiliar with can be added to the project on an on-going basis. To add a term, click on the Naming Conventions tab, and click on the Add symbol. This opens a form which allows a name and a definition to be added to the naming convention list. To modify an existing entry, double click on the try and make the required modifications. Entries can also be deleted from the right-click speed menu.

- Clicking on the Contributors tab opens the Contributors page. To add a contributor, click on the Add symbol to open the Add Contributor form. Contributors can be either a participant, facilitator, or scribe; these reflect the roles that people take in participatory workshops.

Environments

An environment might represent a system operating at a particular time of day, or in a particular physical location. Environments encapsulate visible phenomena such as assets, tasks, personas, and attackers, as well as invisible phenomena, such as goals, vulnerabilities, and threats. Environments may be identified at any time, although these may not become apparent until carrying out contextual inquiry and observing how potential users reason about their context of use.


9.1 Adding a new environment

The screenshot shows a web browser window with the URL `demo.cairis.org`. The CAIRIS navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a Logout button. The breadcrumb trail indicates the current location: Home / Environments / Psychosis.

The 'Add Environment' form contains the following fields:

- Environment:** A text input field containing the value 'Psychosis'.
- Short Code:** A text input field containing the value 'PSY'.
- Description:** A text area containing two paragraphs:
 - The exemplar aims to integrate large existing datasets of serial MRI scans and behaviour data coupled to the NeuroGrid image analysis service into a Grid-based database, test image normalisation techniques, and develop a general ontology for a psychosis databas, for use in multi-centre studies.
 - The exemplar tests capabilities of NeuroGrid to deal with restrospective data, assimilate material into databases, and use of the toolkit for normalisation and analysis.

Below the description field is a table with one row:

	Environment
---	-------------

At the bottom of the form are two buttons: 'Update' (in blue) and 'Cancel' (in grey).

- Select the UX/Environments menu to open the Environments form, and click on the Add button to open the new Environment form.
- Enter the name of the environment, a short code, and a description. The short-code is used to prefix requirement ids associated with an environment.
- If this environment is to be a composite environment, i.e. encompass artifacts of other environments, then click on the Add button the environment table, and select the environment to add.

- It is possible an artifact may appear in multiple environments within a composite environment. It is, therefore, necessary to set duplication properties for composite environments. If the maximise radio button is selected, then the maximal values associated with that artifact will be adopted. This may be the highest likelihood value for a threat, or the highest security property values for an asset. If the override radio button is selected then CAIRIS will ensure that the artifact properties are used for the overriding environment.

Note: Composite environments are an experimental feature and you may get errors when using them.

CHAPTER 10

Assets

Assets are tangible objects of value to stakeholders. By defining an asset in CAIRIS, we implicitly state that this needs to be secured in light of risks which subsequently get defined.

Assets are situated in one or more environments. Security and Privacy properties are associated with each asset for every environment it can be found in. These properties are described below:

Property	Description	Reference
Confidentiality	The property that information is not made available or disclosed to unauthorised individuals, entities, or processes.	ISO/IEC 27001
Integrity	The property of safeguarding the accuracy or completeness of assets.	ISO/IEC 27001
Availability	The property of being accessible and usable on demand by an authorised entity.	ISO/IEC 27001
Accountability	The property that ensures the actions of an entity may be traced uniquely to an entity.	ISO 7498-2
Anonymity	The property that other users or subjects are unable to determine the identity of a user bound to a subject or operation.	Common Criteria Privacy Requirements
Pseudonymity	The property that a set of users and/or subjects are unable to determine the identity of a user bound to a subject or operation, but that this user is still accountable for its action.	Common Criteria Privacy Requirements
Unlinkability	The property that users and/or subjects are unable to determine whether the same user caused certain operations in the system.	Common Criteria Privacy Requirements
Unobservability	The property that users and/or subjects cannot determine whether an operation is being performed.	Common Criteria Privacy Requirements

Each of these properties is associated with the value of None, Low, Medium, or High. The meaning of each of these values can be defined in CAIRIS from the Asset Values dialog; this is available via the Options/Asset values menu.

10.1 Adding, updating, and deleting an asset

CAIRIS Home System Requirements Risk UX Models Options Search Search Logout

Home / Assets / PLC

Summary Criticality Interfaces

Asset PLC **Shortcode** PLC **Type** Systems

Description Programmable Logic Controller.

Significance Compromising PLCs can compromise connected assets, and other assets associated with it.

Tags

+ Environment

Day Night

Definition Associations

	Property	Value	Rationale
+	Integrity	High	Tampering impacts monitoring of water quality.
-	Availability	High	Controls machinery

Update Cancel

- Select the Risks/Assets menu button to open the assets table, and click on the Add button to open a new asset form.
- Enter the name of the asset, a short code, description, and significance. The short-code is used to prefix requirement ids associated with an environment.
- If this asset is deemed critical, click on the Criticality tab, and click on the Critical Asset check-box. A rationale for declaring this asset critical should also be added. By declaring an asset critical, any risk which either threatens or exploits this asset will be maximised until the mitigations render the likelihood of the threat or the severity of the vulnerability inert.
- Click on the Add button in the environment card, and select an environment to situate the asset in. This will add the new environment to the environment tab.
- After ensuring the environment is selected in the environment table, add the security properties to this asset for this environment. Security properties are added by clicking on the Add button in the properties table to open the Choose security property dialog. From this window, a security property, its value its value rationale can be added.
- Click on the Create button to add the new asset.
- Existing assets can be modified by double clicking on the asset in the Risks/Assets table, making the necessary changes, and clicking on the Update button.

- To delete an asset, select the asset to delete in the assets table, and select the Delete button. If any artifacts are dependent on this asset then a modal dialog stating these dependencies are displayed. The user has the option of selecting Yes to remove the asset dependencies and the asset itself, or No to cancel the deletion.

10.2 Asset modelling

Understanding how assets can be associated with each other is a useful means of identifying where the weak links in a prospective architecture might be. CAIRIS supports the association of assets, inconsistency checking between associated assets, and visualisation of asset models.

The CAIRIS asset model is based on UML class models. Asset models can be viewed for each defined environment. As well as explicitly defined asset associations, asset models will also contain associations implicitly defined. For example, if a task has been defined, and this task concerns within an environment contain one or more assets, then the participating persona will be displayed as an actor, and an association between this actor and the asset will be displayed. Additionally, if concern associations have been defined between goals and assets and/or associations then zooming into the model will display these concerns; the concerns are displayed as blue comment elements.

10.2.1 Adding an asset association

- You can add an association between assets by selecting the Risk/Asset Association menu, and clicking on the Add button in the association table.
- In the association form which is opened., set the adornments for the head and tail end of the association. Possible adornment options are Inheritance, Association, Aggregation, and Composition; the semantics for these adornments are based on UML.
- Set the multiplicity (nry) for the head and tail ends of the association. Possible multiplicity options are 1, *, and 1..*.
- Optional role names can also be set at the head or tail end of the association.
- Check the navigation setting for the head and tail ends of the association. By default, this is 0. Setting an end to 1 indicates that an asset at the opposite end of the association has visibility of assets on the end set. This is consistent with navigability semantics in UML class diagrams.
- Select the Create (or Update if modifying an existing association) will add the association to the CAIRIS model.
- You can also add associations between other assets from the environment Associations tab within the Asset form. You can add a new association by clicking on the Add button in the association table to open the association form. From this form, you can add details about the nature of the association between the asset you're working on and another [tail] asset. Once you click on Update, the association will be added to your working object, but won't be committed to the model until you click on the Update/Create button.

Although not possible from the UI, it is possible to add associations between assets directly in a CAIRIS model file without first defining security or privacy properties for the asset in the model file. If you do this, all the security and privacy properties for the asset are set to None and the rationale of `Implicit` is set for each property.

10.2.2 Viewing Asset models

Asset models can be viewed by selecting the Models/Asset menu, and selecting the environment to view the environment for.

By changing the environment name in the environment combo box, the asset model for a different environment can be viewed.

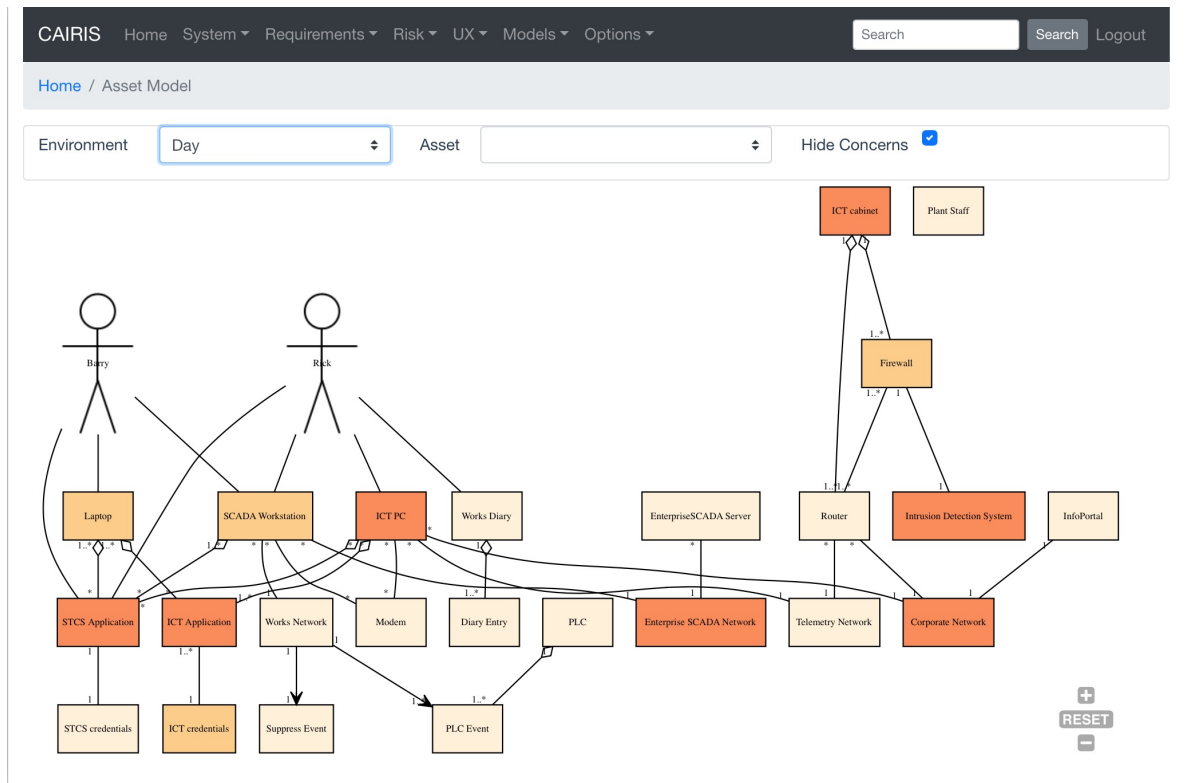
Each asset node is shaded red based on its *attack surface*. This is based on the highest severity value of the vulnerabilities associated with the asset; the higher the value, the darker the shade of red.

The screenshot shows a web browser window with the URL `demo.cairis.org`. The page title is "CAIRIS" and the navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options. A search bar and a "Search" button are also present. The breadcrumb trail is "Home / Asset Associations / Day / ICT PC / STCS Application".

The main form is titled "Asset Associations" and contains the following fields and options:

- Environment:** A dropdown menu with "Day" selected.
- Head:** A dropdown menu with "ICT PC" selected.
- Navigation:** Radio buttons for "1" and "0", with "0" selected.
- Type:** Radio buttons for "Inheritance", "Association", "Aggregation", and "Composition", with "Aggregation" selected.
- Multiplicity:** Radio buttons for "1", "*", and "1..*", with "*" selected.
- Role:** A text input field.
- Role:** A text input field.
- Multiplicity:** Radio buttons for "1", "*", and "1..*", with "*" selected.
- Type:** Radio buttons for "Inheritance", "Association", "Aggregation", and "Composition", with "Association" selected.
- Navigation:** Radio buttons for "1" and "0", with "0" selected.
- Tail:** A dropdown menu with "STCS Application" selected.
- Rationale:** A text input field containing the text "Some STCS applications are installed to control devices on the telemetry network."

At the bottom of the form are two buttons: "Update" and "Cancel".



The model can be filtered by selecting an asset. This will display on the asset, and the other asset model elements immediately associated with it. By default, concern associations are hidden. These are UML comment nodes that indicate elements from other CAIRIS models associated with asset. These concerns can be shown by changing the Hide Concerns combo box value to Yes.

By clicking on a model element, information about that artifact can be viewed.

For details on how to print asset models as SVG files, see [Generating Documentation](#).

10.2.3 Template Assets

You can specify libraries of template assets that you might form the basis of security or architectural patterns.

These can be added, updated, and deleted in much the same way as standard assets, but with two differences:

1. Template assets are not environment specific, so you need to specify the general security properties that need to be protected should this asset be included in a model.
2. You need to first define Access Rights, Surface Types, and Privileges.

10.2.4 Asset modelling with diagrams.net

[diagrams.net](#) (previously known as draw.io) is an easy to use, open source diagramming tool; it can be run either from the browser or from the desktop. diagrams.net has the ability to set shape properties and export to XML and, as a result, asset models created in this tool can, if defined properly, be imported into CAIRIS by following the steps below:

1. Create a new blank diagram in [diagrams.net](#).
2. Setup the CAIRIS asset shape library by going to the File >> Open Library from >> URL menu, and entering the URL `https://cairis.org/stencils/cairis_asset.xml`.

CAIRIS Home System Requirements Risk UX Models Options Search test

Home / Template assets / Widget Processor

Summary Interfaces

Asset
Widget Processor

Shortcode
WPROC

Type
Software

Description
Widget Processor

Significance
Responsible for implementing the widget installation logic

Surface Type
Privileged application

Access Right
trusted

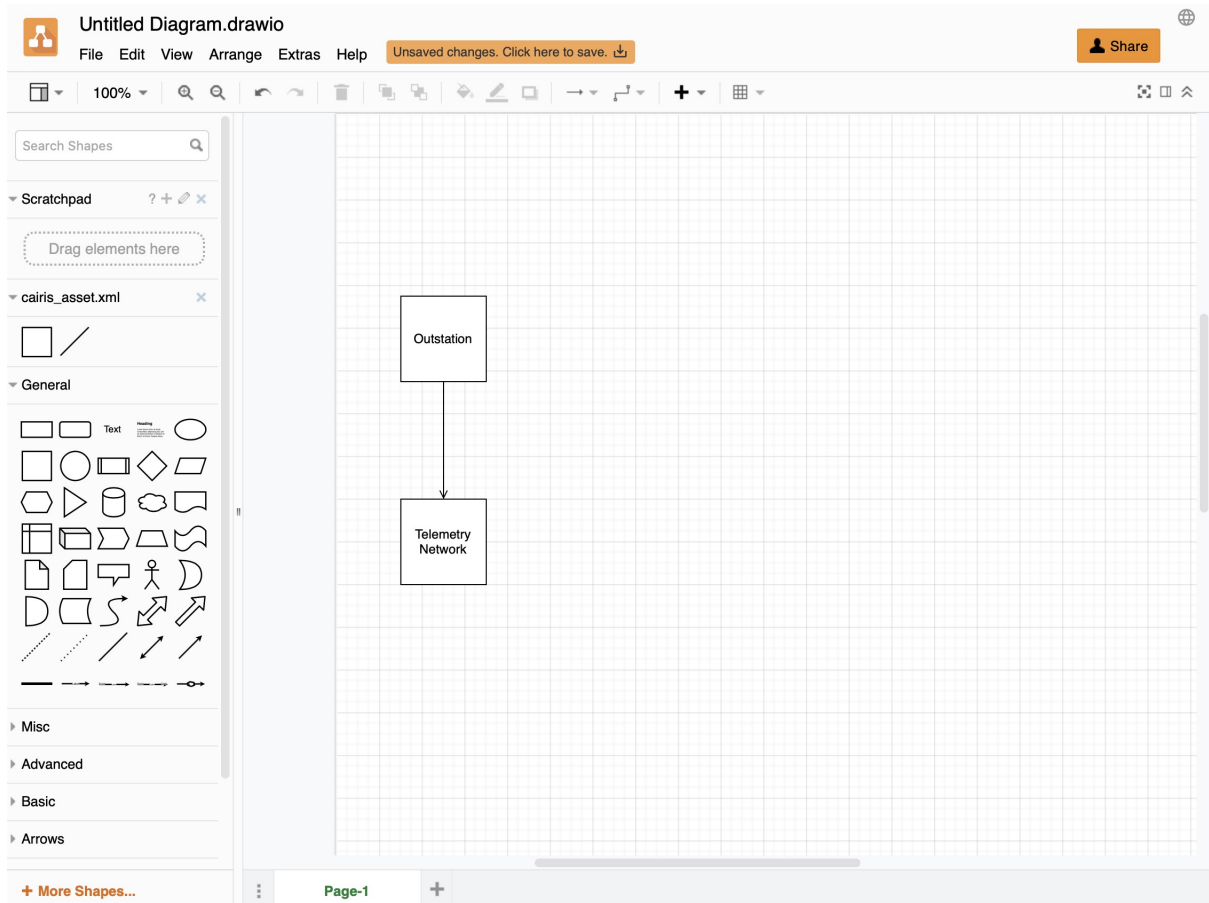
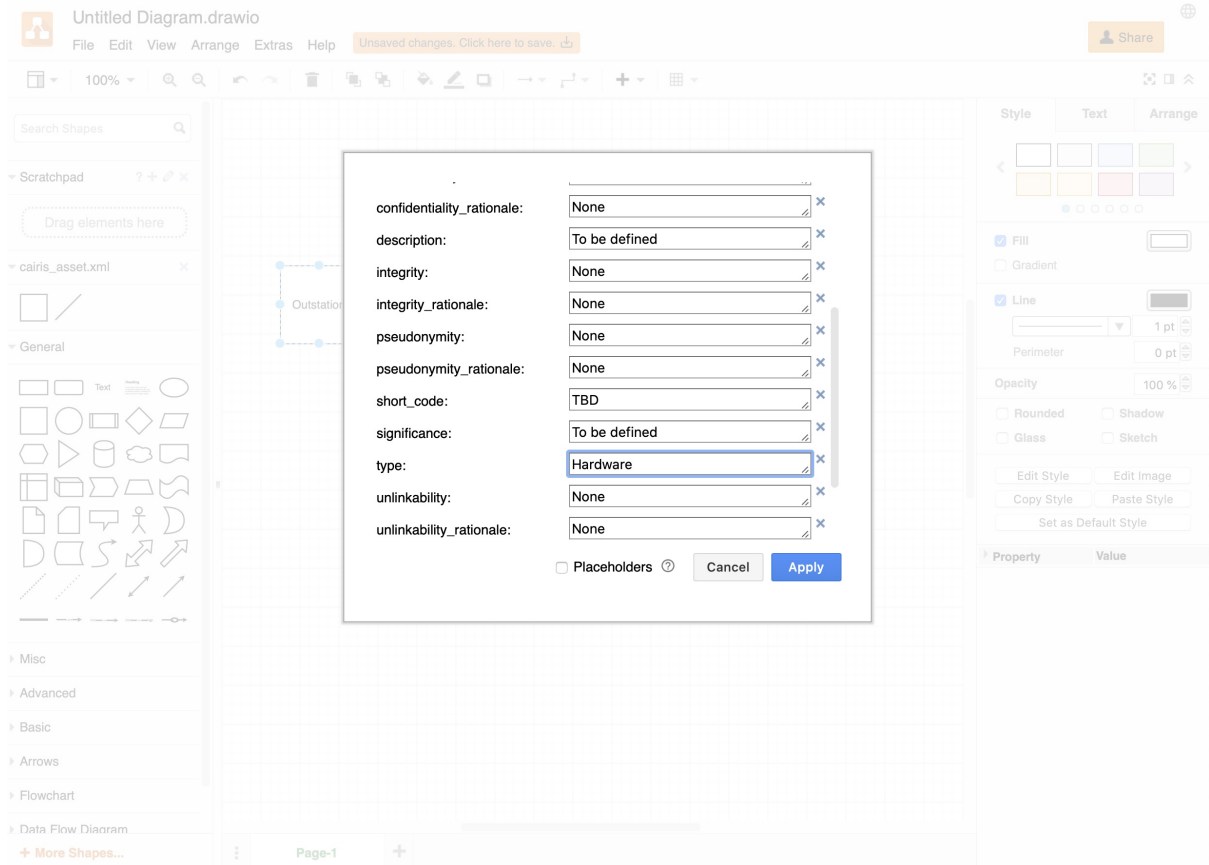
Tags
Enter new tags separated by comma

Property	Value	Rationale
<div>Update Cancel</div>		

- Click on the square (asset) in the `cairis_assets.xml` palette to place an asset on the canvas. Double click on the shape to set its label, which represents the asset name. Hovering the mouse over the asset will display the asset properties as a tool-tip
- Right click on the asset and select *Edit Data* to change the asset properties. When changing the asset, ensure you enter only a permissible value for the type (Hardware, Software, Information, Systems, or People) and the security property values (None, Low, Medium, or High).
- Click on the line (asset association) in the `cairis_assets.xml` palette to place an asset association on the canvas. Change the start of end arrow accordingly based on the nature of the asset association. For example, setting an open arrow on the Telemetry Network asset end of the association indicates navigability from the Outstation to the Telemetry Network. When adding associations between asset, ensure the association line is connected to both assets.
- Once the diagram is ready, select the File >> Export as >> XML... menu option, unclick the Compressed tick box, click on the Export button, and enter the name of the diagram to be exported.
- In CAIRIS, select the System >> Import menu to open the Import form. Select *diagrams.net (Asset Model)* from the Model combo box, click on the File button to choose the exported diagrams.net model to import, and select the environment to import the asset model into.

Assets that don't already exist will be created in CAIRIS, with security properties set for the environment the model is imported into. Assets and associations that already exist will not be overwritten.

Note: We recommend you use the `cairis_asset.xml` shape library when asset modelling, but you could - in theory - use any shape in diagrams.net to model assets. However, you must ensure that you use the Edit Data option to add a `type` property to the shape, which should be set to a valid asset type.



CAIRIS

Home

System

Requirements

Risk

UX

Models

Options

Home / Asset Model

Environment

Asset

Hide Concerns

Refresh

Day

Outstation

Telemetry Network

Outstation

Outstation

Type

Description

Significance

Property

Value

Rationale

Availability

Low

To be defined

Close

Telemetry Network

Outstation

Roles are abstract classes representing human agents; these also encapsulate behaviours and responsibilities. CAIRIS supports 6 types of role:

Role	Description	
Stakeholder	Human agents the system needs to be directly or indirectly designed for.	IRIS Meta-model
Attacker	Human agents behaving maliciously.	IRIS Meta-model
Data Controller	The entity that determines the purposes, conditions and means of the processing of personal data.	GDPR
Data Processor	The entity that processes data on behalf of the Data Controller.	GDPR
Data Subject	A natural person whose personal data is processed by a controller or processor.	GDPR
Machine	Software agents that behave with some level of autonomy.	

11.1 Adding, updating, and deleting a role

- Select the Risk/Roles menu to open the Roles table, and click on the Add button to open the Role form.
- Enter a role name and description, and select the role type.
- Click on the Create button to Add the new role to the CAIRIS database.
- Existing roles can be modified by clicking on the role in the roles table, making the necessary changes, and clicking on the Update button.
- To delete a role, select the role to delete in the roles table, If any artifacts are dependent on this role then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the role dependencies and the role itself, or No to cancel the deletion.

CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

SearchUser ▾

Home / Roles / Certificate Authority

Role

Certificate Authority

Short Code

CA

Type

Stakeholder ▾

Description

Authorises access requests for NeuroGrid and responsible for day-to-day administration.

Create

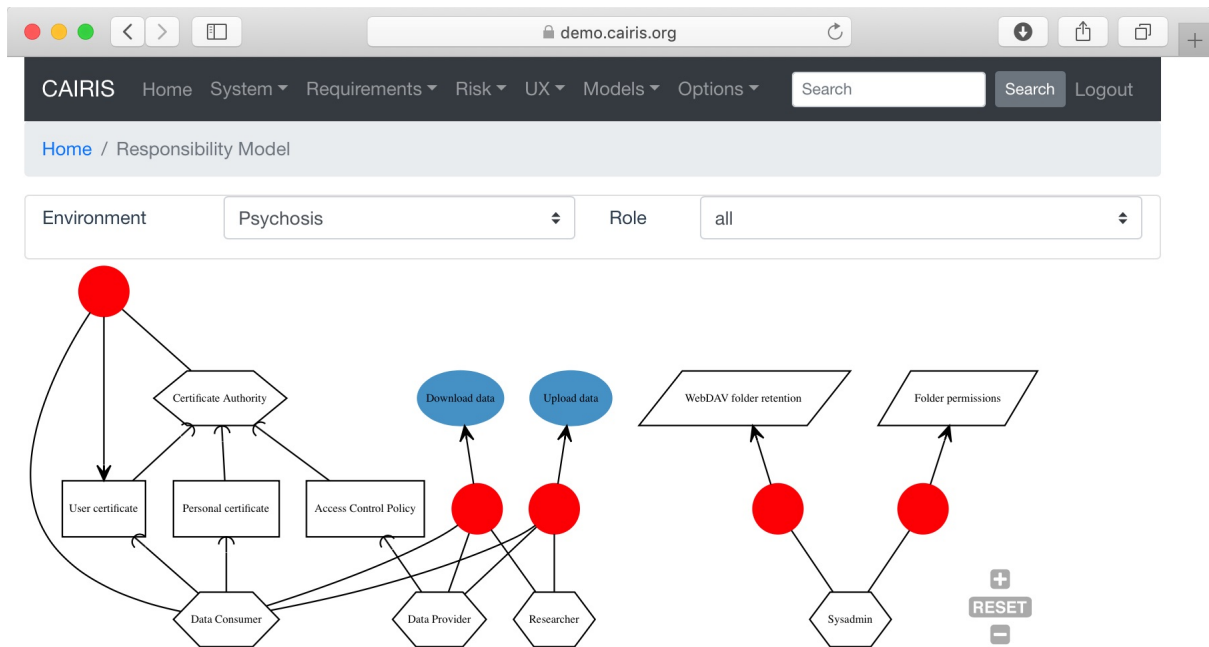
Cancel

11.2 Responsibility modelling

Responsibility models can be viewed by selecting the Models/Responsibility menu option and selecting the environment to view the environment for.

By changing the environment name in the environment combo box, the responsibility model for a different environment can be viewed. By clicking on a model element, information about that artifact can be viewed.

For details on how to print responsibility models as SVG files, see [Generating Documentation](#).



CHAPTER 12

Personas

Personas are specifications of archetypical users that the system needs to directly or indirectly cater for. The system needs to be specified for Primary Personas, but Secondary Personas cannot be ignored as their thoughts or concerns provide insight into potential usability problems.

12.1 Adding, updating, or deleting a persona

The screenshot shows the CAIRIS system interface. The top navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a User dropdown. The breadcrumb trail indicates the current location: Home / Personas / Barry.

The main content area displays the 'Personas' form for a user named Barry. The form includes a photo of Barry and a tabbed interface with the following tabs: Summary, Activities, Attitudes, Aptitudes, Motivations, Skills, Contextual Trust, and Intrinsic Trust. The 'Summary' tab is currently selected, showing a text area with the following content:

Much of Barry's work involves equipment modification and instrument calibration; this work arises from requests from Process, or activities as part of on-going projects to improve plant efficiency. As part of these changes, Barry may need to modify alarms on outstations, and make minor changes to PLCs and HMIs.

Barry is often called out to troubleshoot problems at sites and kiosks within his area of operations in Dorset, which can also involve software changes ranging from simply downloading software to a device to clean up its memory, through to re-generating the software from configuration sheets if no software backup is available.

Below the text area, there is a section for 'Environment' with a '+ Environment' button. Underneath, there are two buttons: 'Day' (selected) and 'Night'. Below these are two tabs: 'Roles' and 'Narrative'. The 'Roles' tab is selected, showing a table with the following data:

Role
Instrument Technician

At the bottom of the form, there are two buttons: 'Update' and 'Cancel'.

- Select the UX/Personas menu to open the table of personas, and click on the Add button to open the new Persona form.

- Enter a persona name and select the persona type.
- If the persona is grounded in assumption-based data, click on the Assumption-based checkbox.
- If you have decided to personalise the persona with a picture, this can be added by clicking on avatar silhouette next to the persona description, and selecting a image to represent the persona. Permitted image types are jpg, png, giff, and bmp.
- Click on the Activities tab and enter the activities carried out by the personas.
- Click on the Attitudes tab and enter the attitudes held by the persona, with respect to the problem domain the system will be situated in.
- Click on the Aptitudes tab and enter the persona's aptitudes, with respect to the problem domain the system will be situated in.
- Click on the Motivations tab and enter the persona's personal motivations.
- Click on the Skills tab and enter the persona's skill-set, with respect to the problem domain the system will be situated in.
- Click on the Contextual Trust tab, and enter information about aspects of this persona with an impact on contextual trust warranting properties.
- Click on the Contextual Trust tab, and enter information about aspects of this persona with an impact on intrinsic trust warranting properties.
- If you have decided to personalise the persona with a picture, this can be added by clicking on avatar box next to the persona properties notebook, to select an image to associated with the persona.
- Click on the Environment card, and click on the Add button to situate the persona in an environment. Selecting an environment from the modal will open up a new folder for information about persona roles, and an environment specific narrative.
- After ensuring the environment is selected in the environment window, click on the Roles tab. Select the Direct User checkbox if the persona is a direct stakeholder with respect to the system being defined, and add roles fulfilled by the persona in the Roles list-box. These roles can be added by clicking on the add button in the role table, or deleted by clicking on the button next to the role to be removed.
- Click on the Narrative tab and enter a narrative describing the persona's relationship with the problem domain or prospective system within the environment, and any environment specific concerns he or she might have.
- Click on the Create button to add the new persona.
- Existing personas can be modified by clicking on the persona in the UX/Personas table, making the necessary changes, and clicking on the Update button.
- To delete a persona, click on the delete button next to persona to be removed in the personas table. If any artifacts are dependent on this persona then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the persona dependencies and the persona itself, or No to cancel the deletion.

12.2 Assured personas with persona characteristics

12.2.1 Overview

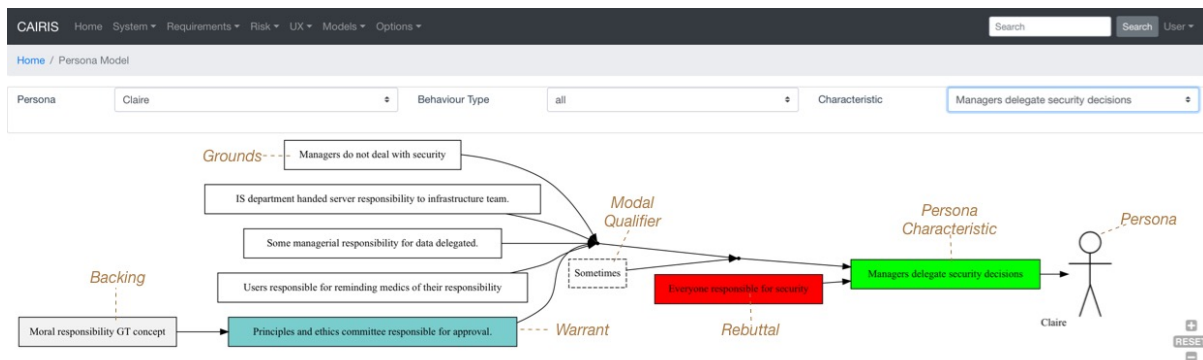
Persona specifications are necessary, but not sufficient for indicating the validity of a persona; you should also describe the basis for each part of the persona specification too. Personas might be created on the basis of some user research. The results of this user research might be coded as a collection of factoids – statements about the data that might be true or false – before the user research makes sense of this data using an activity like affinity diagramming. Clusters of factoids resulting from this exercise form the basis of each aspect of the persona. Normally, however, this data and the results of the analysis are discarded once the persona is created, which

means there is no rationale to justify the persona should questions of clarification of legitimacy be asked about them.

To overcome this problem, CAIRIS supports the creation of *persona characteristics*. These are argumentation models where the *argument* is an individual persona characteristic.

Justifying each characteristic is a one or more *grounds* that provide evidence to support the persona's validity, *warrants* that act as inference rules connecting the grounds to the characteristic, and *rebuttals* that act as counter-arguments for the characteristic. A *model qualifier* is also used to describe the confidence in the validity of the persona characteristic.

This approach for structuring persona characteristic elements is based on Toulmin's model of argumentation¹ and can be visualised in CAIRIS using the persona model, accessible from the Models/Persona menu. As shown in the persona model below, a link can be seen between grounds element and their *backing*, the originating source of the grounds.



12.2.2 Creating persona characteristics

- Select the UX/External Documents menu, and click on the Add button to add information about the source of any assumptions external to CAIRIS. An example of such an *External Document* might be an interview transcript. Alternatively, if assumptions are purely based on your own thoughts and feelings then an External Document can be created to make this explicit. External documents are shown as backing elements in persona models.
- Select the UX/Document References menu, and click on the Add button. Enter a name that summarises the *factoid* that acts as evidence for the persona characteristic. Select the external document from the Document combobox box to indicate the document that the factoid is taken from, and enter details of the person who elicited the assumption in the Contributor text box. Finally, in the Excerpt box, enter the extract of text from the external document from which the factoid is based.
- Select the UX/Persona Characteristics menu, and click on the Add button.
- From the Characteristic folder, enter a definition that summarises the characteristic, and select the Persona and behavioural variable that this characteristic will be associated with. Possible Enter a *Model Qualifier*; this word describes your confidence in the validity of the characteristic. Possible qualifiers might include *always*, *usually*, or *perhaps*.
- In the Grounds table, click on the Add button to add a grounds for the characteristic. Click on the Add button to add a new document reference that acts as grounds. When a document reference is selected, a read-only description of this document reference will also be shown. Clicking Ok will add the new document reference to the grounds list.
- Repeat the above procedure for *Warrants* as appropriate.
- If you wish to add a Rebuttal – a counterargument for the characteristic – then click on the Rebuttals tab and add a rebuttal using the same procedure for Grounds and Warrants.
- Click on the Create button to create the new characteristic.

¹ Toulmin, S. The uses of argument, updated ed. Cambridge University Press, 2003.

CAIRIS Home System Requirements Risk UX Models Options Search Search Logout

Home / Persona characteristics / Managers delegate security decisions

Characteristic GRL Elements

Definition

Managers delegate security decisions

Persona

Claire

Variable

Activities

Modal Qualifier

Sometimes

+ Grounds

- IS department handed server responsibility to infrastructure team.
- Managers do not deal with security
- Some managerial responsibility for data delegated.
- Users responsible for reminding medics of their responsibility

+ Warrant

- Principles and ethics committee responsible for approval.

+ Rebuttal

- Everyone responsible for security

Update Cancel

- Existing characteristics can be modified by double clicking on the characteristics in the persona characteristics table, making the necessary changes, and clicking on the Update button.

12.3 Automating persona characteristic creation

In the ideal world, personas will be created by dedicated teams of research collecting empirical data, working collectively in one place to affinity diagram factoids, and persona characteristics that structure them. In reality, team members might be working individually, remotely, and using open source intelligence or online sources of data. To provide some automation for this activity, we have created some features for offline and collaborative creation of persona characteristics.

12.3.1 Working with persona characteristic workbooks

CAIRIS can generate an Excel workbook that makes it possible to add or update persona characteristic elements.

To export a workbook, select the System/Export menu, click on the *Persona characteristics (Workbook)* radio button, enter the name of the workbook to be created, and click on the Export button. The workbook name should be postfixed with .xlsx.

CAIRIS Home System Requirements Risk UX Models Options Search Search User

Home / Export

Model

☐ Model
 ☐ Model (XML file)
 ☐ GRL
 ☐ Architectural Pattern
 ☐ Security Patterns
 ☒ Persona characteristics (Workbook)
 ☐ User goals (Workbook)

File name

ACME_PC.xlsx

Export Cancel

The generated spreadsheet contains three spreadsheets: External Documents, Document References, and Persona Characteristics.

	A	B	C	D	E
	Name	Authors	Version	Publication Date	Description
1	big security worry GT concept	Shamal Faily	1	August 2010	big security worry GT concept
2	Alarm handling GT concept	Shamal Faily	1	August 2010	Alarm handling GT concept
3	Authorisation restriction GT concept	Shamal Faily	1	August 2010	Authorisation restriction GT concept
4	Business compliance GT concept	Shamal Faily	1	August 2010	Business compliance GT concept
5	Corporate interface GT concept	Shamal Faily	1	August 2010	Corporate interface GT concept
6	Equipment GT concept	Shamal Faily	1	August 2010	Equipment GT concept
7	External agencies GT concept	Shamal Faily	1	August 2010	External agencies GT concept
8	Hardware checks GT concept	Shamal Faily	1	August 2010	Hardware checks GT concept
9	Hazards GT concept	Shamal Faily	1	August 2010	Hazards GT concept
10	ICT non-support GT concept	Shamal Faily	1	August 2010	ICT non-support GT concept
11	InfoSec communication GT concept	Shamal Faily	1	August 2010	InfoSec communication GT concept
12	InfoSec indifference GT concept	Shamal Faily	1	August 2010	InfoSec indifference GT concept
13	Insider knowledge GT concept	Shamal Faily	1	August 2010	Insider knowledge GT concept
14	Interim Systems GT concept	Shamal Faily	1	August 2010	Interim Systems GT concept
15	Island mindset GT concept	Shamal Faily	1	August 2010	Island mindset GT concept
16	Limited support knowledge GT concept	Shamal Faily	1	August 2010	Limited support knowledge GT concept
17	Local standards GT concept	Shamal Faily	1	August 2010	Local standards GT concept
18	Logout policy GT concept	Shamal Faily	1	August 2010	Logout policy GT concept
19	One-man site GT concept	Shamal Faily	1	August 2010	One-man site GT concept
20	Operational contexts GT concept	Shamal Faily	1	August 2010	Operational contexts GT concept
21	Personal Security GT concept	Shamal Faily	1	August 2010	Personal Security GT concept
22	Process impact GT concept	Shamal Faily	1	August 2010	Process impact GT concept
23	Process Innovation GT concept	Shamal Faily	1	August 2010	Process Innovation GT concept
24	Sample collection GT concept	Shamal Faily	1	August 2010	Sample collection GT concept
25	Scope of responsibility GT concept	Shamal Faily	1	August 2010	Scope of responsibility GT concept

Pre-existing external documents will be added to the External Documents sheet. Updating existing values will update the corresponding object when the spreadsheet is uploaded, but changing the name will create a new external document. To add a new external document, add a row to the spreadsheet and complete the name, author, version, publication date, and description fields.

Pre-existing document references will be added to the Document References sheet. Updating existing values will update the corresponding object when the spreadsheet is uploaded, but changing the name will create a new document reference. To add a new document reference, add a row to the spreadsheet and enter the name, select the document (external document), and enter the contributor and excerpt. If external documents are changed, ensure the document fields in the sheet correspond with an external document - either in the spreadsheet or in the upstream CAIRIS model.

AutoSave OFF

ACME_PC

Home Insert Draw Page Layout Formulas Data Review View Developer Tell me

Share Comments

Paste

Calibri (Body) 11

General

Conditional Formatting Format as Table Cell Styles

Insert Delete Format

Sort & Filter Find & Select

Ideas Sensitivity

B2

big security worry GT concept

	A	B	C	D
	Name	External Document	Contributor	Excerpt
1	Periodic national security alerts	big security worry GT concept	Shamali Faily	Regular emails are received about things like the Iraqi Wars and terrorism; it then goes quite for a while until the next event.
2	Annual police meetings	big security worry GT concept	Shamali Faily	Annual security meetings at Stepford when the police come down because it is a key site.
3	Callouts occur for anything other than a basic trip.	Alarm handling GT concept	Shamali Faily	Basic trips can be manually reset but anything more complicated warrants a call-out.
4	Alarms can be overridden with permission	Alarm handling GT concept	Shamali Faily	Alarms can be overridden but only with permission from Process.
5	Some event alarms are annoying rather than informative	Alarm handling GT concept	Shamali Faily	Alarms come through for certain events rather than problems which are annoying.
6	Role restrictions	Authorisation restriction GT concept	Shamali Faily	Depending on what you are authorised to do - certain buttons for things like starting and stopping pumps will be greyed out.
7	Line manager site authorisation	Authorisation restriction GT concept	Shamali Faily	Can only access sites they have been authorised to; permission for authorisation changes need to be sought from the line manager.
8	Work reports are filed	Business compliance GT concept	Shamali Faily	Work reports are filed and sent to ACME monthly.
9	Everything is logged	Business compliance GT concept	Shamali Faily	Everything that happens is logged.
10	Corporate windows logstart policy	Business compliance GT concept	Shamali Faily	Policy for locking Windows down is corporate.
11	Logs are recorded in the Management System	Business compliance GT concept	Shamali Faily	Logs are recorded in the Management System.
12	MS activities during start of day shift.	Business compliance GT concept	Shamali Faily	Management System activities are carried out during the first hours of a day shift.
13	Everything has a MS procedure	Business compliance GT concept	Shamali Faily	Everything has a procedure in the Management System.
14	The works diary is a controlled document.	Business compliance GT concept	Shamali Faily	The works diary is a controlled document.
15	Jobs inputted at the end of a shift	Business compliance GT concept	Shamali Faily	Jobs tend to be input at the end of a shift.
16	MS workaround for intruder alarms.	Business compliance GT concept	Shamali Faily	Lack of police response to intruder alarms led to two-man requirement for alarm response.
17	Tasks entered into AIS are captured by WMS	Corporate interface GT concept	Shamali Faily	Tasks are entered into Asset Job System are captured by Work Management System.
18	Users identified by WMS numbers	Corporate interface GT concept	Shamali Faily	Users are identified by unique WMS numbers.
19	AIS shows current jobs	Corporate interface GT concept	Shamali Faily	AIS indicates how many jobs you have.
20	Contact details on Outlook	Equipment GT concept	Shamali Faily	You can get all the contact details you need from outlook.
21	Communal PC for checking email	Equipment GT concept	Shamali Faily	A communal PC is used for checking mail.
22	Panel PCs around plant can run plant	Equipment GT concept	Shamali Faily	Panel PCs are used to check things while checking on a delivery.
23	HMI passwords	Equipment GT concept	Shamali Faily	HMIs have passwords associated with them.
24	Clients rather than HMIs	Equipment GT concept	Shamali Faily	The new system moves towards clients rather than HMIs.
25	SCADA PCs are site-specific industrial PCs	Equipment GT concept	Shamali Faily	SCADA PCs are site-specific industrial PCs.
26	SCADA authentication via non ACME credentials	Equipment GT concept	Shamali Faily	Uses non ACME passwords to log onto the SCADA and pumping station.
27	Environmental Agency carry out river monitoring	External agencies GT concept	Shamali Faily	The Environmental Agency carry out their own river monitoring.
28	Out-of-spec alarms precipitate TIS call-outs	External agencies GT concept	Shamali Faily	Out of spec alarms are picked up Telemetry Information System; these lead to a technician call out; independent sampling; log checking.
29	Readings are taken from SCADA screens.	Hardware checks GT concept	Shamali Faily	SCADA screens are checked to take readings about chemical dosing; iron levels; pH etc.
30	Hardware is checked daily	Hardware checks GT concept	Shamali Faily	Hardware equipment is checked daily
31	Flooding warning on a sewage site is a flood.	Hazards GT concept	Shamali Faily	Worse case scenario on a sewage site is a flood.
32	Massive and uncontrollable water input	Hazards GT concept	Shamali Faily	It is not like a factory: it is a massive system with tonnes of debris that hits you all in one go.
33	Pumping decisions rain based	Hazards GT concept	Shamali Faily	Decisions to pump are based on the rain forecast.
34	Bad weather can lead to power dips.	Hazards GT concept	Shamali Faily	Bad weather can lead to power dips.
35	Large water tanks conduct electricity	Hazards GT concept	Shamali Faily	Large water tanks conduct electricity
36	Failures often occur when seasons change.	Hazards GT concept	Shamali Faily	Failures often occur when seasons change.
37	Unintimely logout annoyance	ICT non-support GT concept	Shamali Faily	It is annoying to tip downstairs and have to log back in when you come back.
38	Lack of night ICT support	ICT non-support GT concept	Shamali Faily	Why have night support if all you get is a logged reference number; you cannot hand the problem over to someone else.
39	No out-of-office facility	ICT non-support GT concept	Shamali Faily	Managers can set out-of-office messages; plant operators cannot.
40				An account lock-out at night necessitates calling someone out to log them back in to carry out

External Documents Document References Persona Characteristics

100%

React to alarms raised by SCADA						
Characteristic	Persona	Variable	Medial Qualifier	Grounds	Warrant	Rebuttal
React to alarms raised by SCADA	Rick	Activities	Always	Callouts occur for anything other than a basic trip.		
Readings periodically taken from SCADA	Rick	Activities	Always	SCADA screens.	Operators know what they need to do.	
Managers need to authorise what they can and cannot do	Rick	Activities	Always	Role restrictions	Line manager site authorisation	
Area of responsibility is large and unpredictable	Rick	Activities	Generally	Process problems occur daily, Water scheme is geographically big		
Routine varies by time of day	Rick	Activities	Generally	Night shift tasks differ slightly from day, Specific jobs are day shift related	Some plant operations are cheaper at night.	
Scheduled tasks issued by AIS	Rick	Activities	Always	Tasks entered into AIS are captured by WMS	Carries out tasks and records set-point readings	
Process decisions may be weather based.	Rick	Activities	Occasionally	Failures often occur when seasons change, Pumping decisions rain based	Massive and uncontrollable water input	
Processes regularly checked against spec	Rick	Activities	Always	Processes checked against spec, Water entering plant checked against spec	Operators know what they need to do.	
Issues with TIS technicians and the Environmental Agency	Rick	Activities	Occasionally	precipitate TIS call-outs	Anything could happen	
Samples are taken throughout the work and recorded	Rick	Activities	Always	Water is sampled throughout the works	Carries out tasks and records set-point readings	
General work and events written into Work Diary	Rick	Activities	Always	logged	Everything is logged	
Unknown applications are not authorised	Rick	Activities	Always	cannot be run		
Following MS is part of the daily work routine.	Rick	Activities	Always	MS activities during start of day shift.	The works diary is a controlled document.	
Trends regularly checked to spot problems	Rick	Activities	Usually	problems	Trends identify plant problems	
Design decisions built around past experience	Rick	Activities	Often	standards.	Room for improvement	
Standards have been internalised	Rick	Activities	Generally	Standards evolved over years	Works manual is internalised	
Hoping for productivity improvements without adverse impact	Rick	Activities	Hopefully	HMI usability improvements, Quick and automatic startup	Few SCADA to EnterpriseSCADA process changes	
Frequently control plan outside the control room	Rick	Activities	Usually	Plant can be operated away from the control room if necessary.	Process problems occur daily.	
Aware of hacking warnings but scenarios are extreme	Rick	Attitudes	Generally	Extreme scenarios, Hacker aware	Hacking indifference	
ICT support is contextual	Rick	Attitudes	Often	Lack of ICT support, Unintended layout annoyance		
Logon policies should be built around security needs.	Rick	Attitudes	Always	Shift based logouts, Short auto logout	Physical and login security	
Standard migration subject to resources	Rick	Attitudes	Probably	Migration pending resources	Hacking indifference	
Thieves do not care about their impact	Rick	Attitudes	Often	Copper theft, Vandalism assessed and mitigated		
Accidents happen but hacking scenarios are extreme.	Rick	Attitudes	Generally	Extreme scenarios	Hacking indifference	
Personal safety is an infection hygiene factor	Rick	Attitudes	Often	Low worker vulnerability, Personal threats	InfoSec communicates irrelevant	
Concern about EnterpriseSCADA uncertainty	Rick	Attitudes	Generally	Bespoke systems harder to understand and support, Uncertainty about EnterpriseSCADA ability to support general operations.	Process problems occur daily.	
Cannot see how big security problems relate to infosec	Rick	Attitudes	Occasionally	Corporate will insecurity, Physical and login security	Periodic national security alerts.	
Communications	Rick	Attitudes	Usually	No alone out-of-hours visits	MS workload around intruder alarms.	
Personal safety is more than just compliance	Rick	Attitudes	Probably	Hacking indifference, Infection only from engineers	Stand-alone SCADA	
SCADA isolation makes hacking unlikely	Rick	Motivations		Records piece things together, Status board communicates problems	Anything could happen	
Need to communicate and log to be ready for anything	Rick	Motivations	Presumably	Experience contributes to asset management decisions, Trouble-shooting by known task problems.	Checking certain things is a discipline, Site kept running at optimum efficiency	
Event logging helps efficiency	Rick	Motivations	Usually	Copper theft, Thieves steal anything	Personal threats	
Feel unsafe challenging intruders	Rick	Motivations				

Pre-existing persona characteristics will be added to the Persona Characteristics sheet. Updating existing values will update the corresponding object when the spreadsheet is uploaded, but changing the name will create a new persona characteristic. To add a new persona characteristic, add a row to the spreadsheet and enter the characteristic name, corresponding persona name, select the behavioural variable, enter the modal qualifier, and grounds, warrant, and rebuttal. The grounds, warrant, and rebuttal cells take a comma separated list of document reference names. You should ensure the named persona exists in the upstream CAIRIS model, and the names of grounds, warrants, and rebuttals correspond with document reference values in the Document References spreadsheet or the upstream CAIRIS model.

The screenshot shows the CAIRIS web application interface. At the top is a navigation bar with links: CAIRIS, Home, System, Requirements, Risk, UX, Models, Options, a search bar, and a User dropdown. Below this is a breadcrumb trail: Home / Import. The main content area is a light blue box containing an 'Import' dialog. Inside the dialog, there is a 'Model' dropdown menu currently showing 'Persona characteristics (Workbook)'. Below that is a 'File' section with a 'Choose File' button and the text 'ACME_PC.xlsx'. At the bottom of the dialog are two buttons: 'Import' (in blue) and 'Cancel' (in grey).

To import a workbook, select the System/Import menu, select the *Persona characteristics (Workbook)* radio button, choose the name of the workbook to be uploaded, and click on the Import button.

Please note that removing rows from any of the spreadsheets does not remove the corresponding object in the CAIRIS model; these should be removed directly in CAIRIS.

12.3.2 Persona Helper

The **Persona Helper** is a Chrome Extension that can be used to automatically create document references from highlighted text on a web page open in Chrome. This might be useful when eliciting factoids from website.

Once the extension has been installed, you need to connect to your CAIRIS server before use. You can do this right clicking on the CAIRIS extension icon in Chrome and selecting 'Connect to CAIRIS'. A dialog will open that will ask for the CAIRIS server URL, before a pop-up appears that allows you to login to your CAIRIS server.

By default, any document references created will be added to the CAIRIS default database, but you can change this using the 'Change CAIRIS database' menu option.

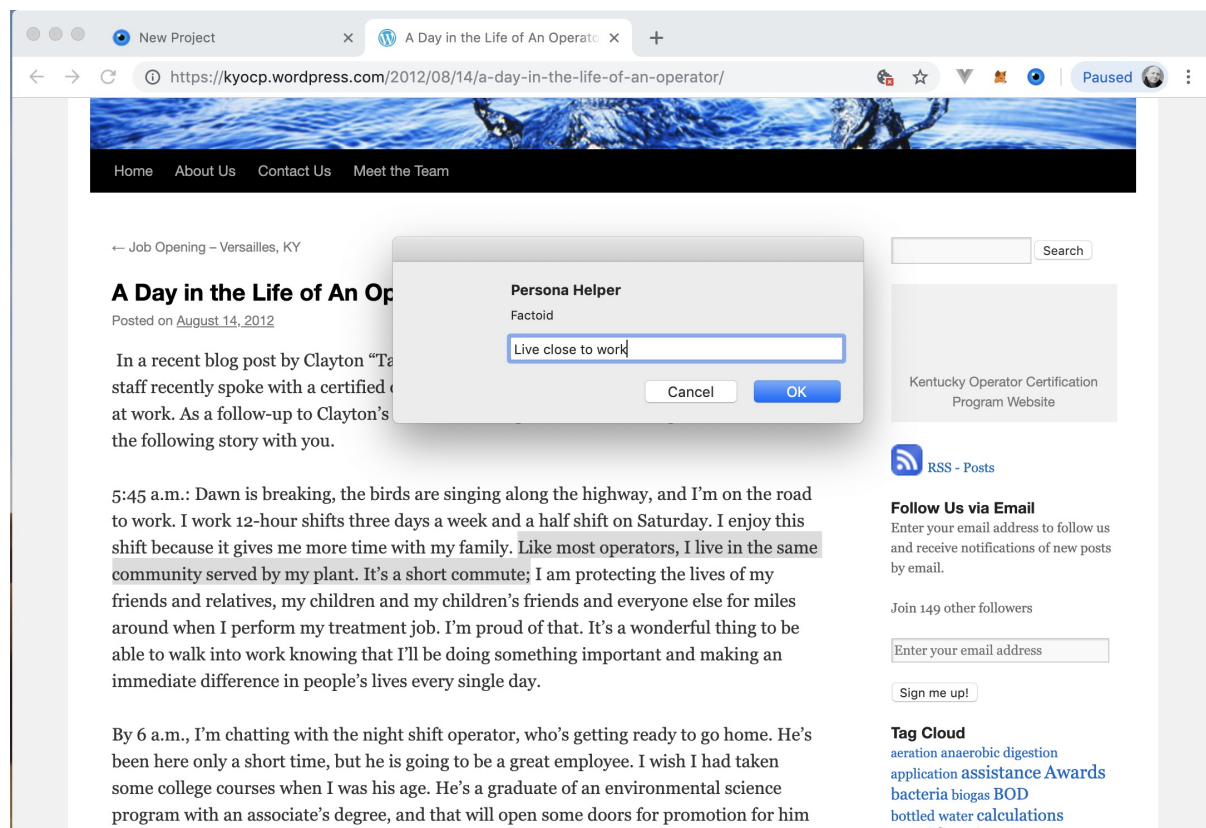
From the extension menu, it also possible to set the *Author* and *Contributor* values. Author is the author of the data source; this will become the author of the external document from which factoids from a website can be drawn. Contributor is the person eliciting the factoid itself. It is ok to set both author and contributor as the same person, but – if different people are responsible for different data sources – you might want to change the author value each time you draw from factoids from a different webpage. If these values have not been previously defined, you will be prompted to provide them the first time you elicit a factoid.

To elicit a factoid, you need to highlight text on a website and click on the CAIRIS extension icon. This will open a dialog that will allow a factoid to be created for the associated text. For example, let's consider we want to build a persona for a water treatment plant operator, and we find a **day in the life of a plant operator** from which we want to elicit factoids.

We find some text that indicates that operators live close to work, so we highlight the relevant text, create some text that describes the factoid (because just because one person in a blog post indicates that plant operators live close to work doesn't mean that most plant operators actually *do* live close to work), and click on Ok to add the factoid.

If we look in CAIRIS, we will see the corresponding document reference as indicated above.

Please remember that CAIRIS is sensitive to reserved characters so, when naming factoids, these should avoided. Colons in factoid names or names of external documents is known to cause particular problems when generating



CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
test

Home / Document references / Live close to work

Name

Live close to work

External Document

A Day in the Life of An Operator | KY OCP

Contributor

SF

Excerpt

Like most operators, I live in the same community served by my plant. It's a short commute;

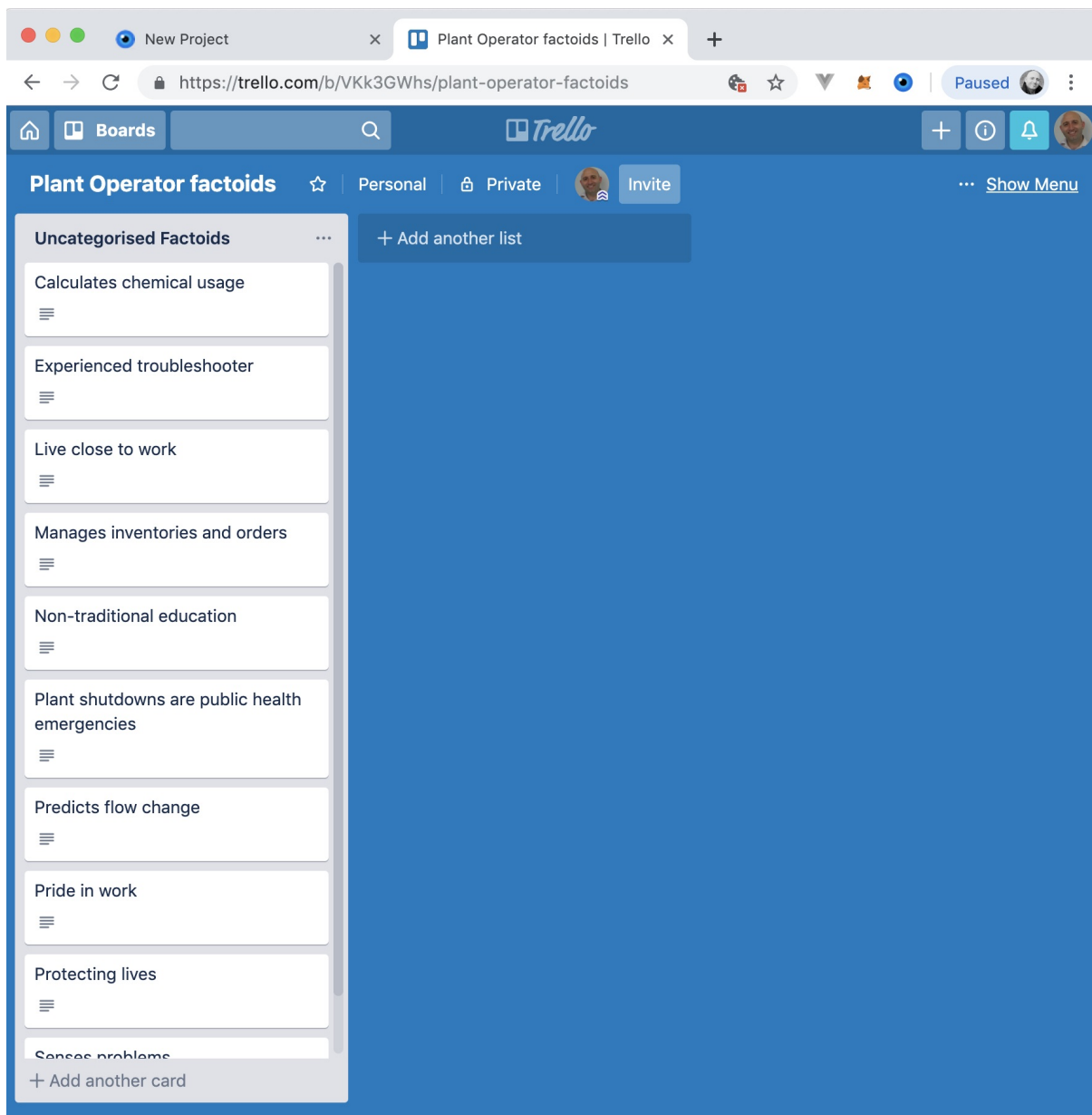
Update
Cancel

persona models.

12.3.3 Online affinity diagramming with Trello

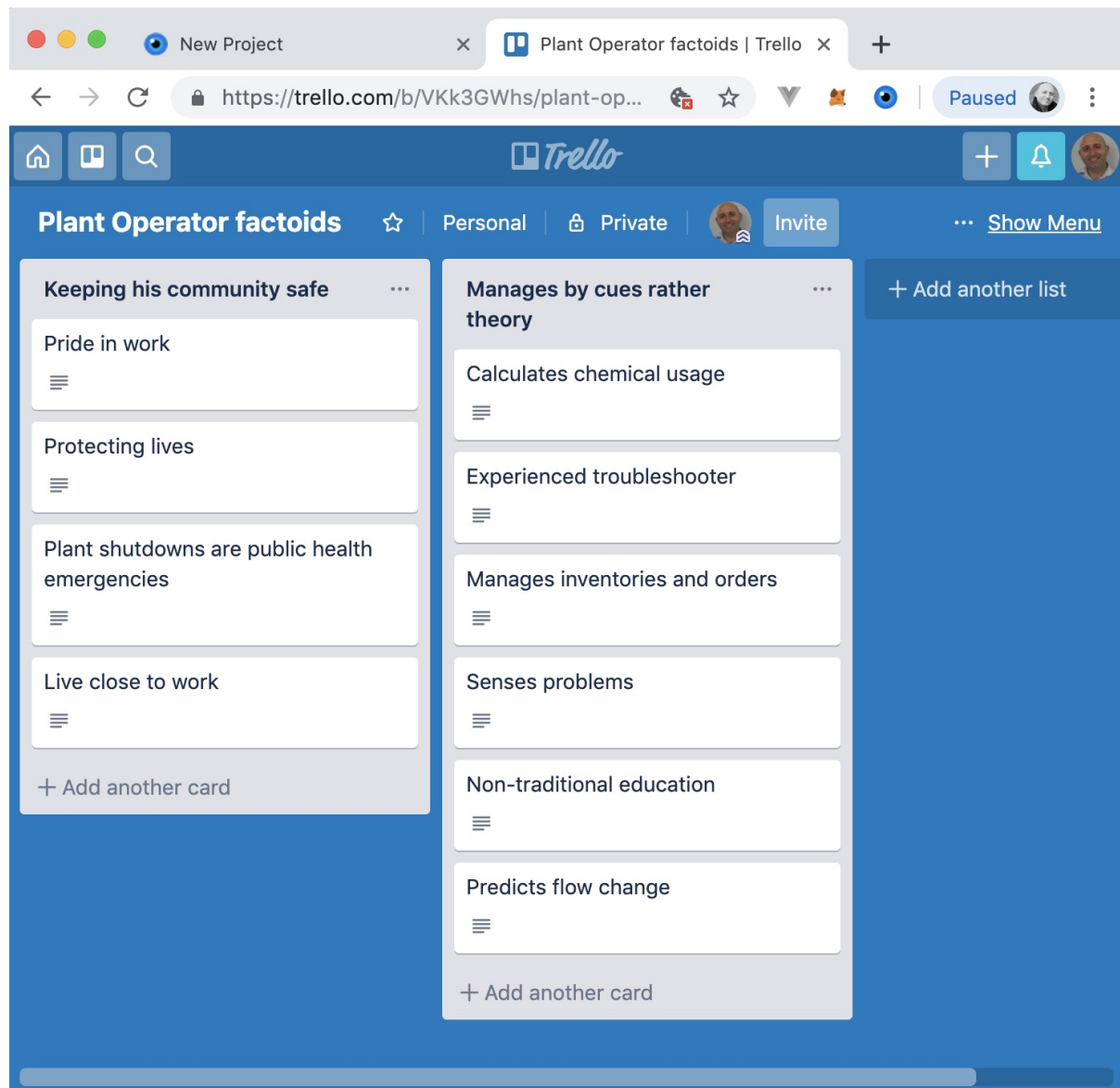
Trello is a collaborative, web-based list manager. It is a popular tool for sharing and collaboratative working on lists, where lists contain cards. Because the relationship between cards and lists is analogous to the relationship between factoids and affinity groups, we can use Trello for online affinity diagramming too. Moreover, because factoids and affinity groups are also analogous with document references and persona characteristics then, using some simple annotations, we can also use lists and cards to represent persona characteristics and their grounds/warrants/rebuttals too.

We can export all the document references in a currently open CAIRIS model by selecting the System/Export to Trello menu. From here, you should enter a Trello board name. This will be created for you once you click on Export, and the document references will be exported as cards to an *Uncategorised Factoids* list in your Trello account as indicated below. If you have not already logged into Trello, you will also be prompted to do this on clicking Export.



As you affinity diagram, each list will represent an affinity group. From the sample of factoids elicited, there seems to be an affinity group around factoids indicating that the plant operator is protective of his community. There is

also another group indicating that the operators relies on cues and experience rather than traditional education and theory.



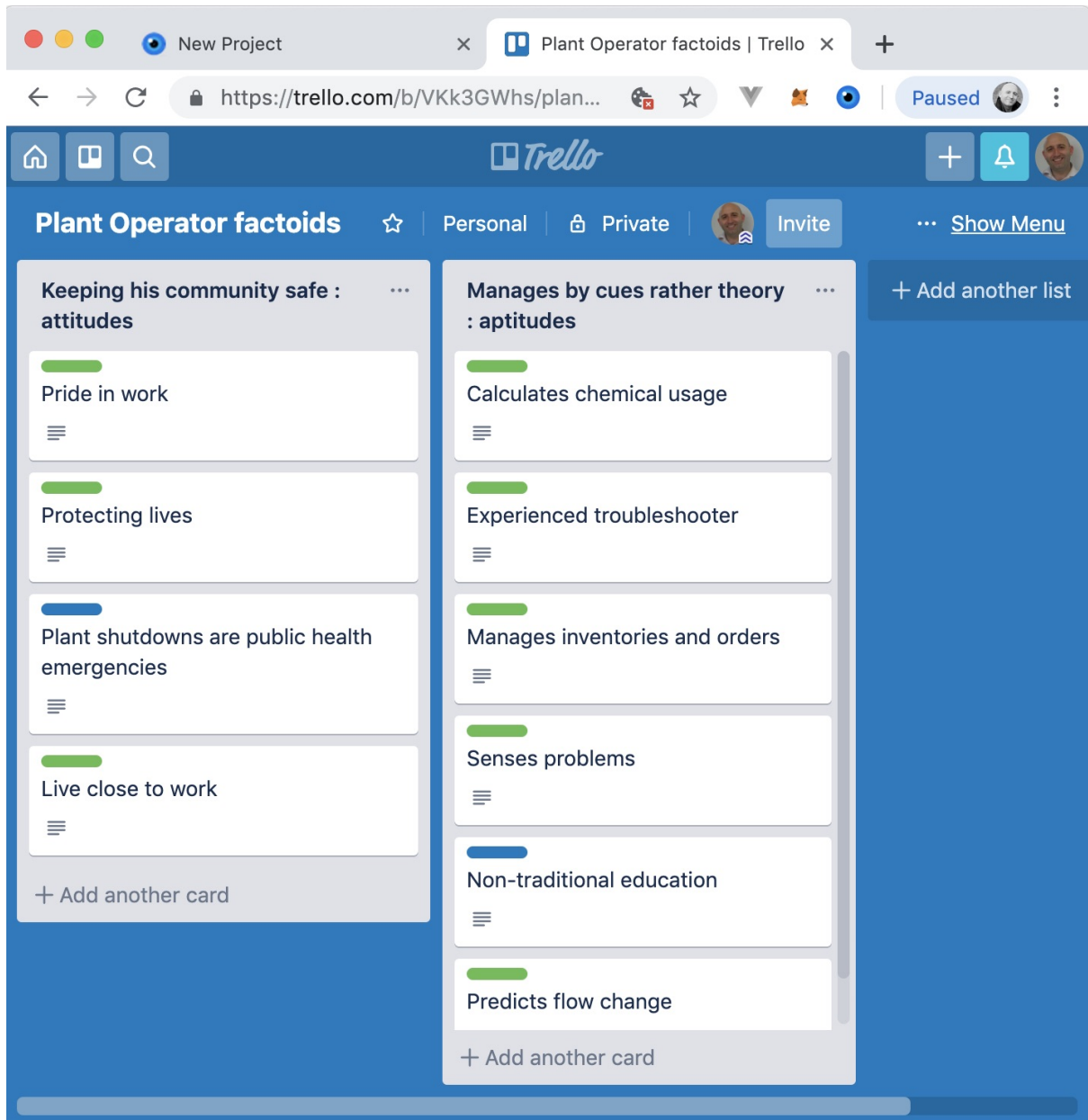
To turn each of these affinity groups into persona characteristics, we first need to indicate whether each factoid represents a groups, warrant or rebuttal. To do this, we click on each card, click on the Label and select either a *grounds*, *warrant* or *rebuttal* label. If you have not imported document references from CAIRIS, these labels won't be automatically created, but you can add them manually.

The final step entails indicating the behavioural variable associated with the persona characteristic. You can do this by postfixing the list name with a colon and the name of the variable. Permissible variable names are: activities, aptitudes, attitudes, motivations, skills, intrinsic and contextual. Please note, you can only associate a persona characteristic with a single behavioural variable.

When you're ready, you can import this Trello board back into CAIRIS. However, before you can do you need to create a persona object for each board you import. For example, an empty persona called Bob will be created to represent a water treatment plant operator.

To import the board, return to CAIRIS and select the System / Import from Trello menu. From this form, you should select the Trello board you are exporting, select the persona associated with the persona characteristics that will be generated, and click on Import.

The persona characteristic generated will, as shown above, have a modal qualifier set to *Perhaps*. This should be updated based on your confidence in the characteristic.



CAIRIS
Home
System ▾
Requirements ▾
Risk ▾
UX ▾
Models ▾
Options ▾

test ▾

Home /
Persona characteristics /
Manages by cues rather theory

Characteristic
GRL Elements

Definition

Persona	Variable	Modal Qualifier
<input type="text" value="Bob"/>	<input type="text" value="Aptitudes"/>	<input type="text" value="Perhaps"/>
<div> + Grounds </div> <div> - Calculates chemical usage </div> <div> - Experienced troubleshooter </div> <div> - Manages inventories and orders </div> <div> - Predicts flow change </div> <div> - Senses problems </div>	<div> + Warrant </div> <div> - Non-traditional education </div>	<div> + Rebuttal </div>

If you have exported your document references from CAIRIS then each ground/warrant/rebuttal document reference will be associated with its appropriate external document. However, if you have created the cards manually in Trello then CAIRIS will create new document references for each card, and an *Unknown* external document to indicate that, at the time of import, the factoids were of uncertain origin. If you know the origin of the factoids, you can create external documents to represent these origins, and re-associate the document references accordingly.

Armed with your persona characteristics, you can now write narrative text in your persona corresponding with these characteristics. In doing so, you may surface possible assumptions or ambiguity. For example, the above persona characteristic seems to suggest a non-traditional education is the basis for managing by cues but this assumption might not be warranted, particularly as all the factoids come from a single source. As such, this could trigger a return to the affinity diagrams or the weakening of the modal qualifier to indicate reduced confidence.

Note: Due to how Trello's client.js file works, you may get errors connecting to Trello during Trello exports and imports. You can get around this by raising an issue in CAIRIS to get your CAIRIS server's URL recognized as an allowable origin. You can also get around this by running the CAIRIS UI in debug mode. To do this by (i) clone the [CAIRIS UI GitHub repository](#), (ii) Follow the instructions in the README to setup the CAIRIS UI project by running *yarn install* in the root directory, (iii) setup the indicated .env.development file as indicated, (iv) running *yarn run serve* to locally run the CAIRIS UI, (v) point your web browser to <http://localhost:8080>.

Tasks model the work carried out by one or more personas. This work is described in environment-specific narrative scenarios, which illustrate how the system is used to augment the work activity.

13.1 Adding, updating, or deleting a task

CAIRIS Home System Requirements Risk UX Models Options Search Search Logout

Home / Tasks / Broken instrument alarm

Task Broken instrument alarm **Short Code** BIA **Author** Anon

Objective Fix flatlined trend

Tags

+ Environment

Day **Night**

Narrative **Participants** **Dependencies** **Consequences** **Benefits** **Concerns**

	Persona	Duration	Frequency	Demands	Goal Conflict
+	Rick	Minutes	Hours or more	Low	Low

Update **Cancel**

- Click on the UX/Tasks menu to open the Tasks table, and click on the Add button to open the Task form.

- Enter a task name, short code, author, and the objective of carrying out the task.
- Click on the Add button in the environment card, and select an environment to situate the task in. This will add the new environment to the environment list.
- In the Narrative folder, enter the task scenario. This narrative should describe how the persona (or personas) carry out the task to achieve the pre-defined objective.
- In the Dependencies folder, enter any dependencies needing to hold before this task can take place.
- In the Consequences folder, enter any consequences (positive or negative) associated with this task.
- In the Benefits folder, enter the value that completing this task will bring.

The screenshot shows the CAIRIS web application interface. A modal dialog titled "Update Task Participation" is open, allowing users to configure task participation parameters. The background shows the "Broken instrument alarm" task page with various tabs and input fields.

Update Task Participation Dialog:

- Persona:** Rick
- Duration:** ☒ Seconds ☒ Minutes ☐ Hours or longer
- Frequency:** ☒ Hours or more ☐ Daily-Weekly ☐ Monthly or less
- Demands:** ☐ None ☒ Low ☐ Medium ☐ High
- Goal Conflict:** ☐ None ☒ Low ☐ Medium ☐ High

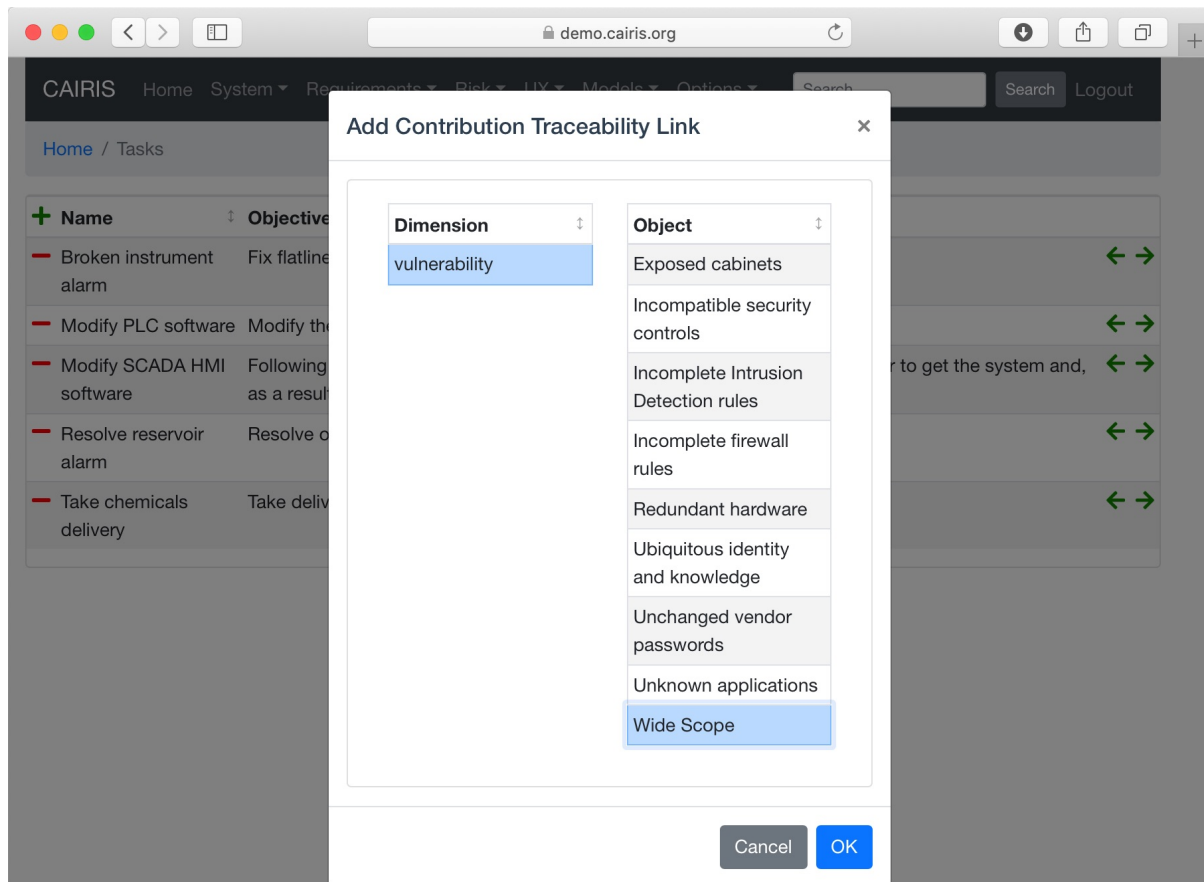
Background Task Page (Broken instrument alarm):

- Task:** Broken instrument alarm
- Objective:** Fix flatlined trend
- Tags:** (empty)
- Environment:** + Environment
- Day/Night:** - Day - Night
- Narrative/Participants:** Narrative Participants
- Participants Table:**

+	Persona	Dura
-	Rick	Minu

- In the Participants folder, click on the Add button to associate a persona with this task. In the Participating Persona form, select the person, the task duration (seconds, minutes, hours or longer), frequency (hourly or more, daily-weekly, monthly or less), demands (none, low, medium, high), and goal conflict (none, low, medium, high). The values for low, medium, and high should be agreed with participants before hand.
- If any aspect of the task concerns one or more assets, then these can be added to the concern list. Adding an asset concern causes a concern comment to be associated to the asset in the asset model. If the task concerns an association between assets, the association can be added by clicking on the Concern Association tab and adding the source and target assets and association multiplicity to the concern association list. In the asset model, this association is displayed and a concern comment is associated to each asset in the association.
- Click on the Create button to add the new task.
- Existing tasks can be modified by clicking on the task in the Tasks table, making the necessary changes, and clicking on the Update button.
- To delete a task, select the task to delete in the Tasks dialog box, and click the Delete button. If any artifacts are dependent on this task then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the task dependencies and the task itself, or No to cancel the deletion.

13.2 Task traceability



Tasks can be manually traced to certain artifacts via the Tasks table. A task may contribute to a vulnerability, or be supported by a requirement or use case. To add a traceability link, right click on the task name, click on the left or right arrows next to the task name to open a Support or Contribution traceability modal respectively. From this editor, select the object on the right hand side of the editor to trace to and click the Add button to add this link.

Manual traceability links can be removed by selecting the Options/Traceability menu option, to open the Traceability Relations form. In this form, manual traceability relations be removed from specific environments.

13.3 Visualising tasks

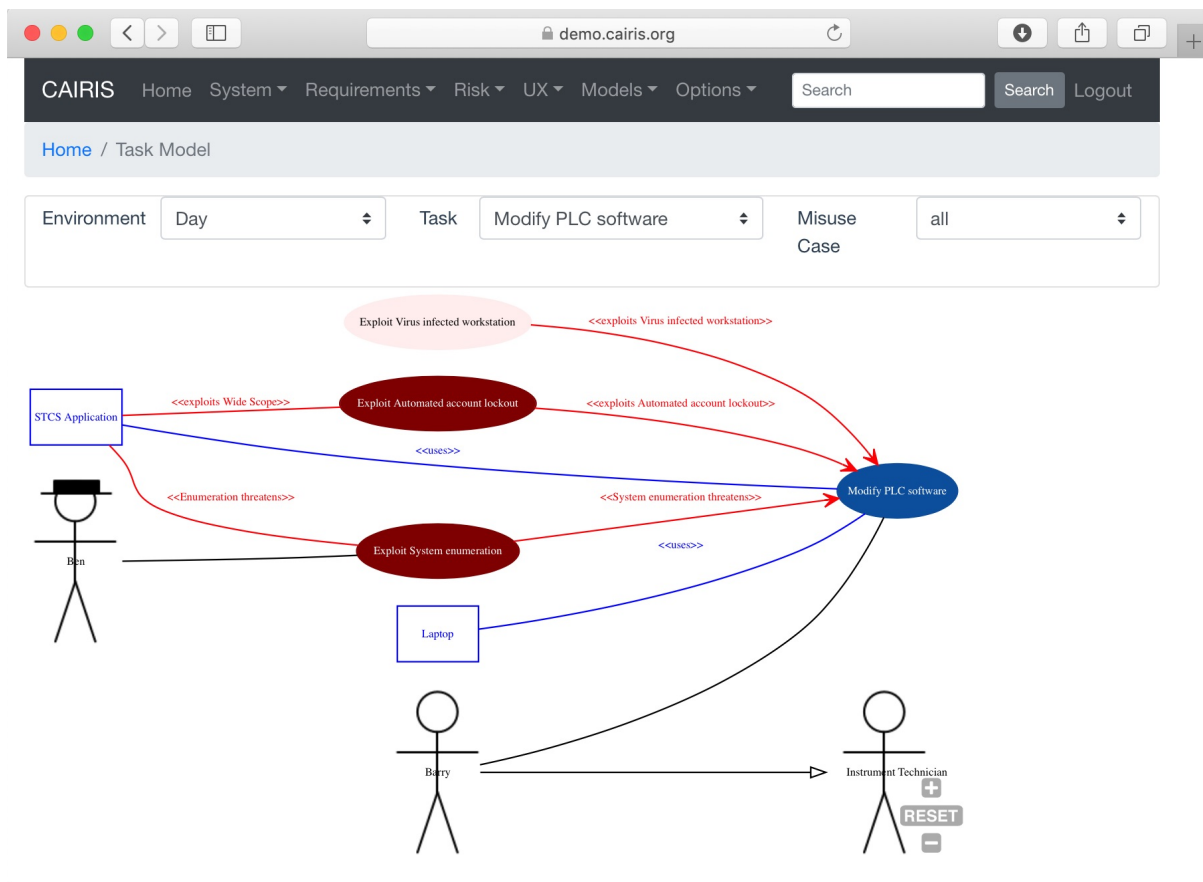
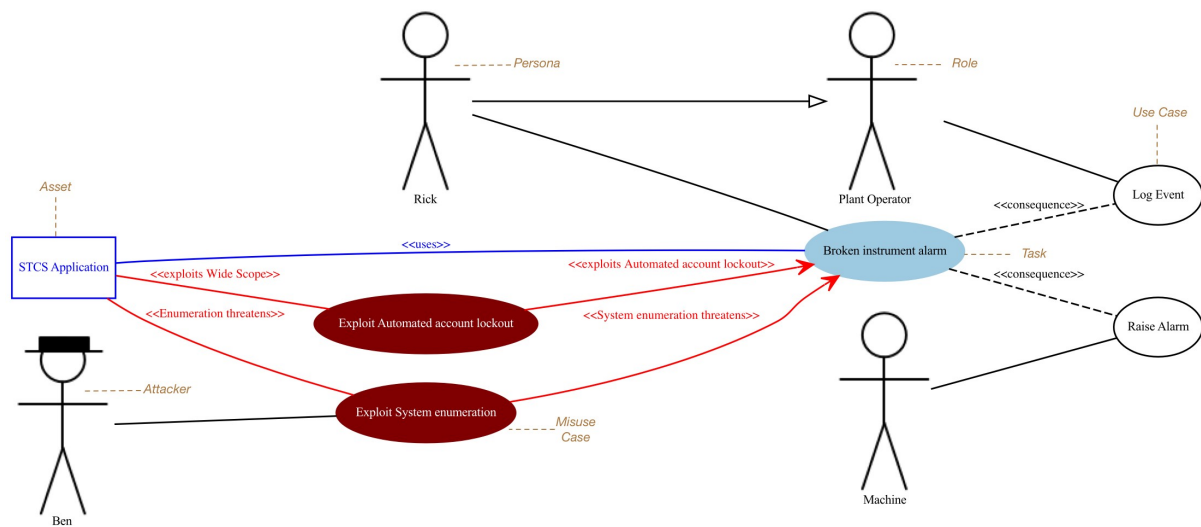
Task models show the contribution that behavioural concepts in security and usability can have on each other. These models are centred around tasks, show the personas that interact with them, and indicate how threats or vulnerabilities might impact them. These models also show the assets used in the tasks or threatened/exploited by misuse cases. If traceability associations have been added between tasks and use cases, then these links are also shown. Finally, if use case actors are also roles associated with personas in visible tasks, then the relationship between the roles and personas is also shown. This is useful when putting use cases and their actors in context in tasks.

Task models can be viewed by selecting the Models/Task menu, and selecting the environment to view the model for.

By changing the environment name in the environment combo box, the task model for a different environment can be viewed. The model can also be filtered by task or misuse case name.

By clicking on a model element, information about that artifact can be viewed.

For details on how to print task models as SVG files, see [Generating Documentation](#).



Misusability Cases

Misusability Cases are scenarios which describe how design decisions may lead to usability problems subsequently leading to system misuse. These can be useful if you feel some aspect of a design might be open to exploitation, and you need to make a case for how the rationale design makes this possible.

14.1 Creating concept references

To create the evidence that forms the basis of your misusability case, you need to create one or more concept references. These can be based on personas, requirements, use cases, or other tasks. The functionality for creating, updating, or deleting concept references is accessible from the UX / Concept References menu, and procedures for working with concept references are very similar to those used for working with document references.

14.2 Creating the skeleton scenario

You need to create a task to encapsulate the misusability case scenario (or scenarios if these are specific to environment).

14.3 Creating task characteristics

Task characteristics form the basis of the argumentation model behind each misusability case. The functionality for creating, updating, or deleting task characteristics is accessible from the UX / Task Characteristics menu, and procedures for working with task characteristics are very similar to those used for working with persona characteristics.

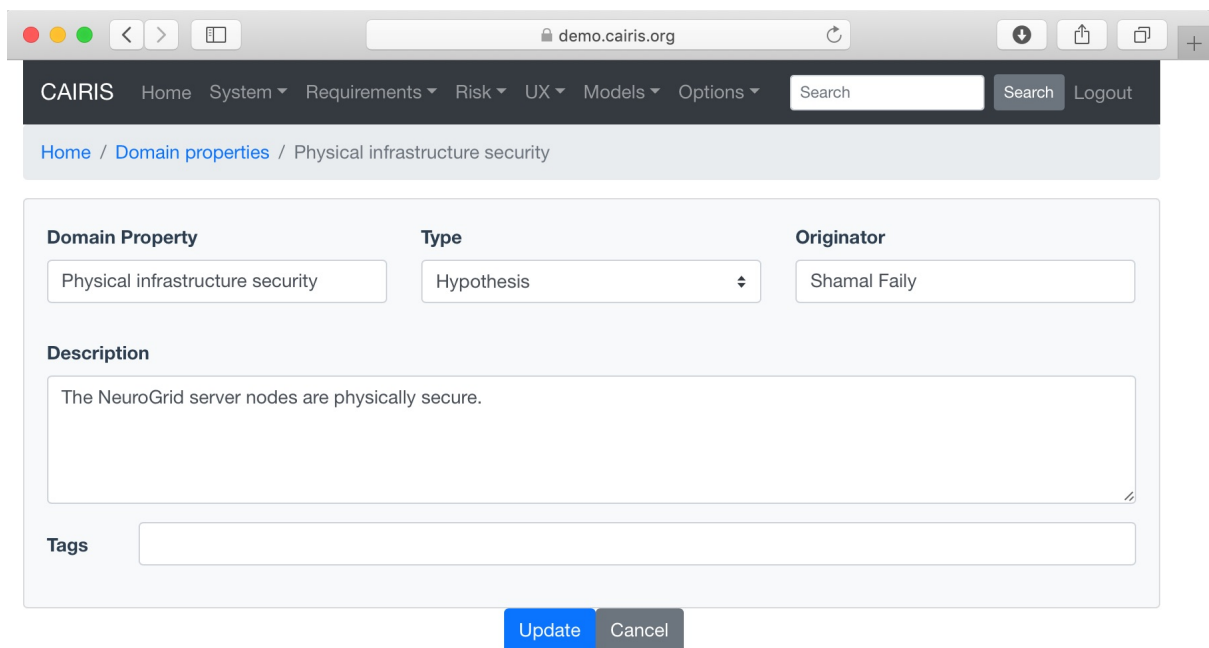
14.4 View misusability case models

You can visualise misusability cases and their supporting argumentation models by selecting the Models / Misusability menu option, and selecting a misusability case to view.

Domain Properties

Domain Properties are descriptive properties about the statement world. Domain Properties may be either hypothesis or invariants.

15.1 Adding, updating, and deleting a domain property



The screenshot shows a web browser window with the URL `demo.cairis.org`. The CAIRIS navigation bar is visible at the top, with a search bar and a 'Logout' button. The breadcrumb trail indicates the current location: `Home / Domain properties / Physical infrastructure security`.

The 'Domain Property' form is displayed with the following fields:

- Domain Property:** Physical infrastructure security
- Type:** Hypothesis (selected from a dropdown menu)
- Originator:** Shamal Faily
- Description:** The NeuroGrid server nodes are physically secure.
- Tags:** (empty input field)

At the bottom of the form are two buttons: 'Update' (highlighted in blue) and 'Cancel'.

- Click on the Requirements/Domain Properties menu to open the Domain Properties table, and click on the Add button to open the Domain Property form.
- Enter a domain property name, description, and select the type of domain property from the type combo box.
- Click on the Create button to add the new domain property.

- Existing domain properties can be modified by clicking on the domain property name in the Domain Properties table, making the necessary changes, and clicking on the Update button.

Goals, Requirements, and Obstacles

In CAIRIS, a requirements specification is analogous to a safety case. In a safety case, a system is only considered safe if its safety goals have been satisfied. In a similar manner, requirements are leaf nodes in a goal tree and satisfying stakeholder needs is only possible if the high-level goals – stipulated by stakeholders – can be satisfied.

We define goals as prescriptive statements of system intent that are achievable by one or more agents. Goals can be refined to requirements, which are achievable by only agent. Goals and requirements may also be operationalised as tasks. Alternatively, we may decide to specify tasks and ask what goals or requirements need to hold in order that a given task can be completed successfully.

To satisfy a goal, one or more sub-goals may need to be satisfied; satisfaction may require satisfying a conjunction of sub-goals, i.e. several AND goals, or a disjunction of sub-goals, i.e. several OR goals.

Goals or requirements may be *obstructed* by obstacles, which are conditions representing undesired behaviour; these prevent an associated goal from being achieved. By progressively refining obstacles, we can obtain the origin of some undesired behaviour; this may be reflected as a vulnerability or a threat, and contribute to risk analysis.

16.1 Adding, updating, and deleting a goal

- Click on the Requirements/Goals button to open the Goals table. As the above figure illustrates, next to goal name is the current *status* for the goal. If a goal is defined as OK, then this goal is refined by a requirement, or by one or more goals. Goals with the status *to refine* have yet to be refined or operationalised. Goals with the status *Check* have been refined by one or more obstacle, and these should be examined to find a root threat or vulnerability.
- Click on the Add button to open the Goal form, and enter the name of the goal.
- Click on the Add button in the environment card, and select an environment to situate the goal in. This will add the new environment to the environment tab.
- In the Definition folder, enter the goal definition, and select the goal category and priority. Possible goal categories are: Achieve, Maintain, Avoid, Improve, Increase, Maximise, and Minimise. Possible priority values are Low, Medium, and High. Enter the fit criteria which must hold for this goal to be satisfied, and any issues or comments relating to this goal.
- If this goal refines a parent goal, click on the Goals tab, click on Add button in the goals table to to open the Add Goal Refinement form. In this form, select the Goal from the Type combo box, and select the Sub-goal, refinement type, and an Alternate value. Possible refinement types are: and, or, conflict, responsible, obstruct, and resolve. The alternative value (Yes or No) indicates whether or not this goal affords a goal-tree

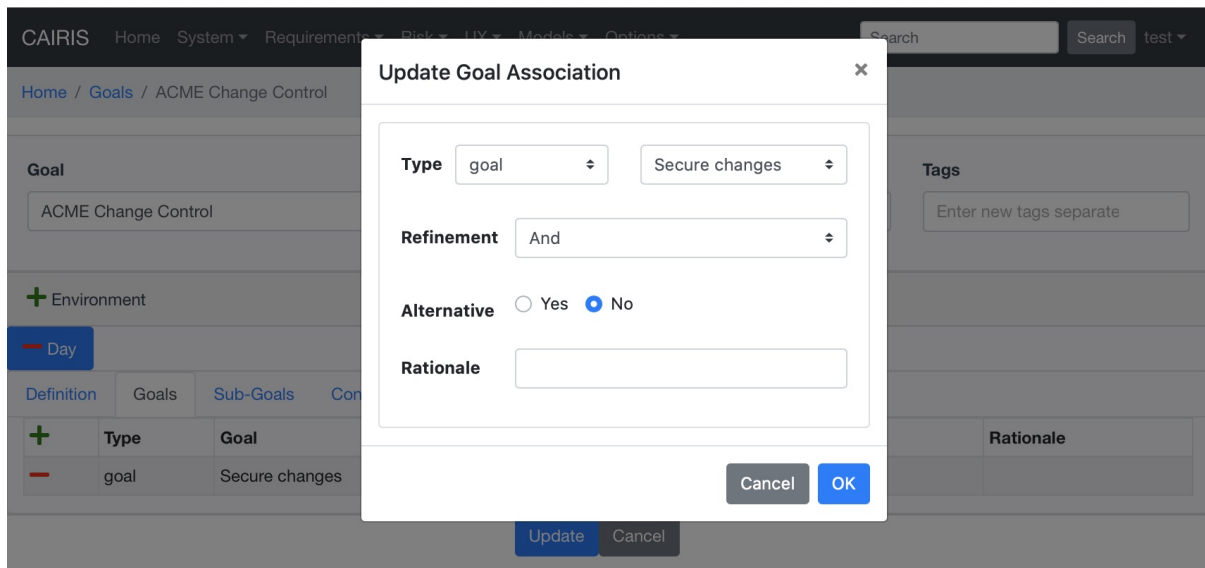
The screenshot shows the CAIRIS web application interface. The top navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a Logout button. The breadcrumb trail indicates the user is in the 'Goals' section. Below the navigation bar, a table lists various goals with columns for Name, Originator, and Status. Each row has a red minus icon on the left. The goals listed are: Access Control (green), Access revocation (red), ACME Change Control (red), Active Directory Network Services (green), Active Directory software deployment (black), Anomalous credentials reset (red), Anti-virus (green), Asset Inventory (green), Authorised removable media (black), Backup hardware (red), Backup logs (red), and Backup media (red).

Name	Originator	Status
Access Control	Anon	green
Access revocation	Anon	red
ACME Change Control	Anon	red
Active Directory Network Services	Anon	green
Active Directory software deployment	Anon	black
Anomalous credentials reset	Anon	red
Anti-virus	Anon	green
Asset Inventory	Anon	green
Authorised removable media	Anon	black
Backup hardware	Anon	red
Backup logs	Anon	red
Backup media	Anon	red

The screenshot shows the 'ACME Change Control' goal details page. The top navigation bar is the same as the previous screenshot. The breadcrumb trail is 'Home / Goals / ACME Change Control'. The form contains the following fields:

- Goal:** ACME Change Control
- Originator:** Anon
- Tags:** Enter new tags separa
- Environment:** + Environment
- Day:** - Day
- Definition:** Goals Sub-Goals Concerns
- Category:** Maintain
- Priority:** Low Medium High (High is selected)
- Definition:** Existing ACME Change Control procedures to co-ordinate RFCs shall be followed
- Fit Criterion:** None
- Issue:** Standard 7.1(n)

At the bottom of the form are 'Update' and 'Cancel' buttons.



for an alternate possibility for satisfying the parent goal. It is also possible to enter a rationale for this goal refinement in the refinement text box. Clicking on Update will add the refinement association to memory, but this will not be committed to the database until the goal is added or updated.

- If this goal refines to sub-goals already specified, Click on the Sub-Goals tab and add a goal refinement association as described in the previous step. A goal may refine to artifacts other than goals, specifically tasks, requirements, obstacles, and domain properties.
- Goal refinements can also be specified independently of goal creation or modification via the Requirements / KAOS Associations menu.
- If any aspect of the goal concerns one or more assets, then these can be added by clicking on the Concerns folder and adding the asset/s to the concern list. Adding an asset concern causes a concern comment to be associated to the asset in the asset model. If the goal concerns an association between assets, the association can be added by clicking on the Concern Association tab and adding the source and target assets and association multiplicity to the concern association list. In the asset model, this association is displayed and a concern comment is associated to each asset in the association.
- Click on the Create button to add the new goal.
- Existing goals can be modified by clicking on the goal name in the Goals table, making the necessary changes, and clicking on the Update button.
- To delete a goal, select the goal to delete in the Goals table, and select the Delete button. If any artifacts are dependent on this goal then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the goal dependencies and the goal itself, or No to cancel the deletion.

16.2 Goal Modelling

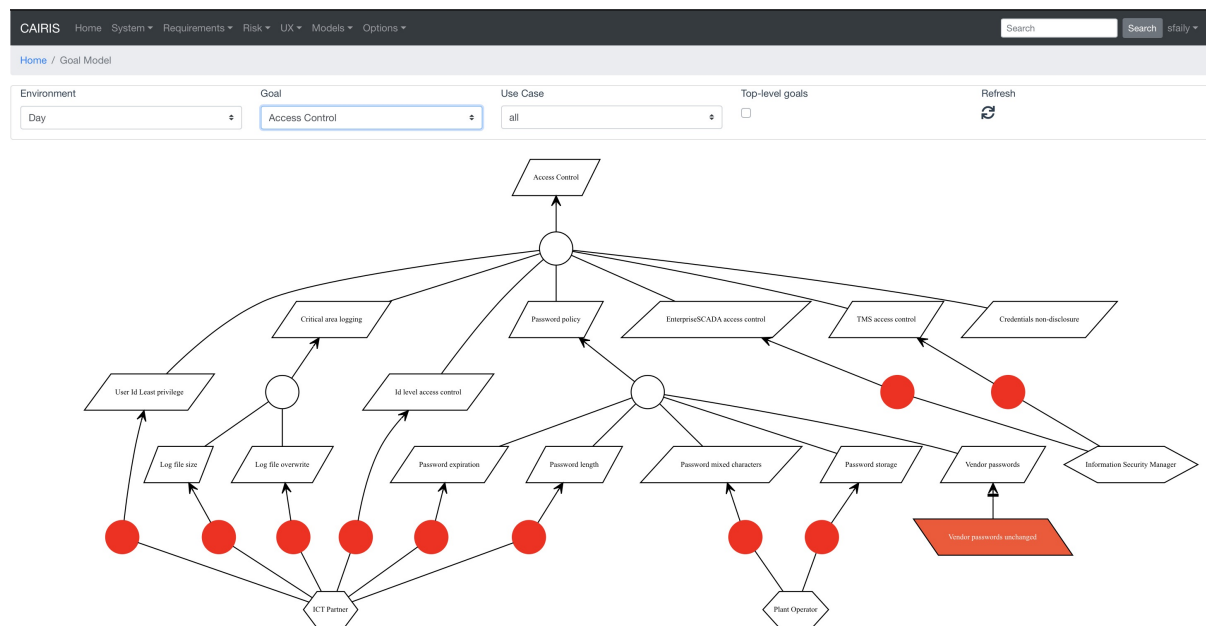
Goal models can be viewed by clicking on the Models/Goal menu option, and selecting the environment to view the environment for.

By changing the environment name in the environment combo box, the goal model for a different environment can be viewed.

By clicking on a model element, information about that artifact can be viewed.

Goal models can also be filtered by goal. Applying a filter causes the selected goal to be displayed as the root goal. Consequently, goals are only displayed if they are direct or indirect leafs of the filtered goal.

For details on how to print goal models as SVG files, see [Generating Documentation](#).



16.2.1 Template Goals

You can specify libraries of template goals that you might form the basis of architectural patterns.

These can be added, updated, and deleted in much the same way as standard goals.

16.3 Adding, updating, and deleting an obstacle

- Click on the Requirements/Obstacle menu to open the Obstacles table box, and click on the Add button to open the Obstacle dform.
- Enter the name of the obstacle, and click on the Add button in the environment card, and select an environment to situate the obstacle in. This will add the new environment to the environment list.
- In the Definition page, enter the obstacle definition, and select the obstacle category. Possible obstacle categories are: Confidentiality Threat, Integrity Threat, Availability Threat, Accountability Threat, Vulnerability, Duration, Frequency, Demands, and Goal Support.
- Enter a probability value (if known), together with a rationale statement justifying the value. When set, probability values need to be between 0 and 1, e.g. 0.2.
- Like goals, obstacle refinements can be added via the Goals and Sub-Goals tabs.
- If any aspect of the obstacle concerns one or more assets, then these can be added by clicking on the Concerns add and adding the asset/s to the concern list. Adding an asset concern causes a concern comment to be associated to the asset in the asset model.
- Click on the Create button to add the new obstacle.
- Existing obstacles can be modified by selecting the obstacle in the Obstacles table, making the necessary changes, and clicking on the Update button.
- To delete an obstacle, select the obstacle to delete in the Obstacles table, and select the Delete button. If any artifacts are dependent on this obstacle then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the obstacle dependencies and the obstacle itself, or No to cancel the deletion.

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
Logout

Home / Template goals / API control

Name

Definition

Rationale

+	Concern
-	Access Requestor
+	Responsibility

Update
Cancel

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
test

Home / Obstacles / Certificate sharing

Obstacle

Originator

Tags

+ Environment

- Psychosis
- Stroke

Definition
Obstacles
Sub-Obstacles

Category

Definition

Probability

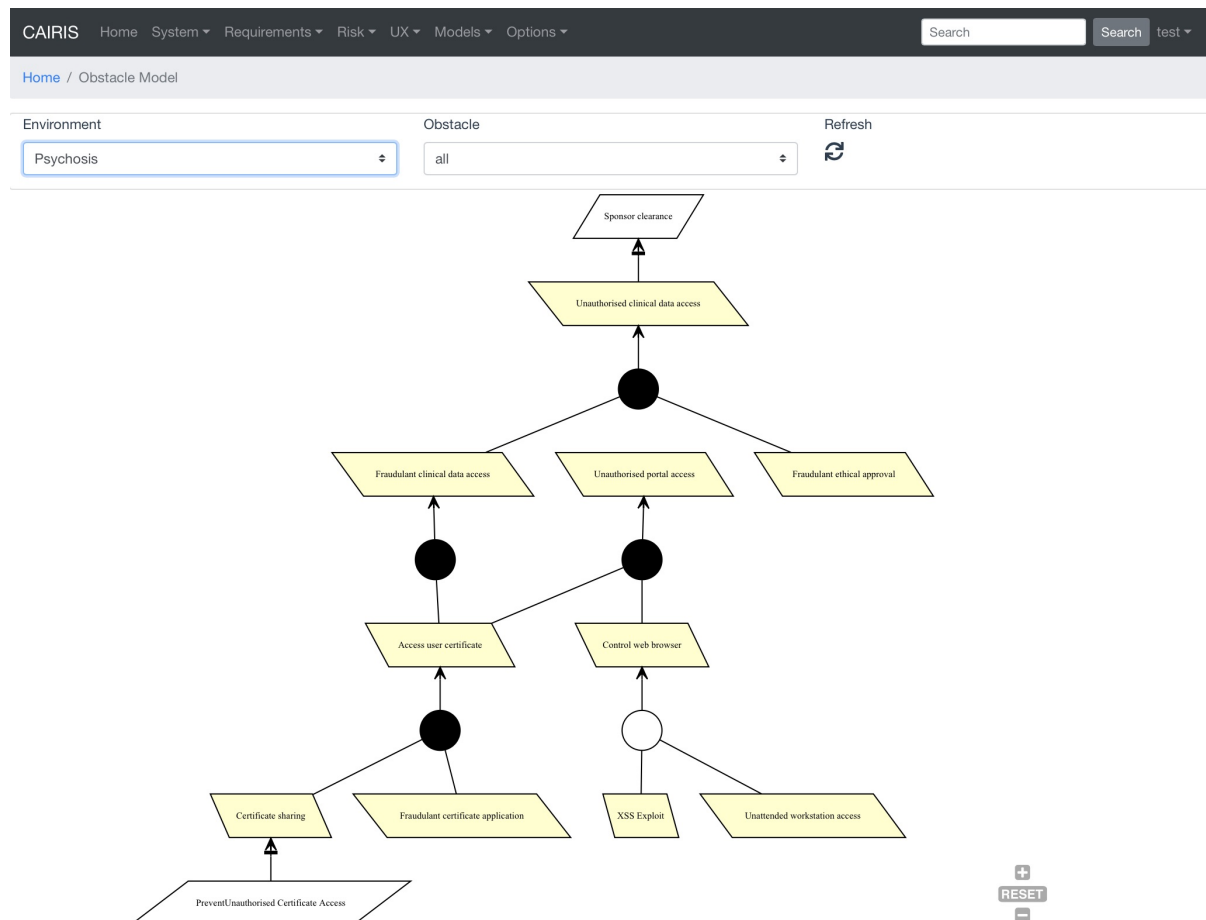
Reason

+	Concern
-	Personal certificate

Update
Cancel

16.4 Obstacle Modelling

Obstacle models can be viewed by clicking on the Models/Obstacle menu button, and selecting the environment to view the environment for.



In many ways, the obstacle model is very similar to the goal model. The main differences are goal filtering is not possible, only the obstacle tree is displayed, and obstacles refine to obstacles, as opposed to goals.

A KAOS obstacle model can be seen as a goal-driven form of a *Fault Tree*. However, unlike fault trees, obstacle modelling is closely tied to other artifacts such as previous knowledge about attacks and information about the attackers that might carry these out.

Where useful statistical data about possible attacks exists, this information can help us predict the likelihood of particular obstacles being satisfied. When a probability value is specified in obstacles for this likelihood then a rationale statement also needs to be provided to justify it. This is necessary because, when attack patterns are imported into a CAIRIS model, it may not be immediately obvious that the obstacle or the obstacle model arose from them. By proving this justification, we have some way of understanding the thinking that motivated this value. Based on these values, we can evaluate the probability of a particular cut of an obstacle tree based on the same equations used to evaluate the faults in a fault tree. For example, for an obstacle O with leaf goals $O1$ and $O2$, the probability of $O1$ where $O1$ and $O2$ are AND-refinements is $O1 \times O2$; where $O1$ and $O2$ are OR-refinements then the probability is $O1 + O2$.

Obstacles are coloured with a shade of red based on the probability set when defining the obstacle. The probability can be a real number between 0 and 1, where the default value is 0.

For details on how to print obstacle models as SVG files, see [Generating Documentation](#).

16.5 Adding, updating, and deleting requirements

The screenshot shows the CAIRIS web interface for adding or updating requirements. The top navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar. The breadcrumb trail indicates the current path: Home / Requirements / Anonymisation guidelines.

The form itself is divided into several sections:

- Name:** A text input field containing "Anonymisation guidelines".
- Type:** A dropdown menu currently set to "Functional".
- Domain:** Radio buttons for "Asset" (selected) and "Environment", followed by a dropdown menu set to "Clinical data".
- Specification:** A text area containing the text: "Anonymisation guidelines shall comply with the MRC guidelines for secure data handling and anonymisation."
- Fit Criterion:** A text area currently set to "None".
- Priority:** Radio buttons for "1" (selected), "2", and "3".
- Rationale:** A text area currently set to "None".
- Originator:** A text area containing "Interview data".

At the bottom of the form are two buttons: "Update" (highlighted in blue) and "Cancel".

Requirements are accessible by selecting the Requirements/Requirements menu option. Each requirement is associated with an asset, or an environment. Requirements associated with assets may specify the asset, constrain the asset, or reference it in some way. Requirements associated with an environment are considered transient, and remain associated with an environment only until appropriate assets are identified.

- To add a requirement, click on the Add button in the requirements table.
- Enter the requirement description, rationale, fit criterion, and originator in the appropriate cells, select the priority (1,2, 3), and the requirement type (Functional, Data, Look and Feel, Usability, Performance, Operational, Maintainability, Portability, Security, Cultural and Political, and Legal).
- When the attributes have been entered, click on the Create button to add the requirement.
- When a requirement has been added or update, the asset/environment filter will be updated based on the asset or environment the new/updated requirement is associated with.
- In the requirements table, a requirement can be deleting by clicking on the delete button.

16.6 Visualising Requirements Quality using Chernoff Faces

Requirements quality is automatically scored based on requirements completeness, the presence of an imperative phrase, and ambiguity.

These are displayed using cartoon *Chernoff Faces*. Eye-brow shape indicates the completeness of a given requirement. If no text is found in certain fields, or phrases like *TBC*, *none*, or *not defined* are present, the completeness score is marked down accordingly, and the eye-brows convey a negative mood.

The eye shape indicates whether or not an imperative phrase exists in the requirement description. If such a phrase exists then the eyes become vertically elongated. The mouth indicates the presence of weak or fuzzy phrases, such as *mostly*, *appropriate*, *normal*, or *adequate*; the presence of these phrases turn the smile into a frown.

Access Control Policy

Access sponsor



Dataset policy



Sponsor clearance



Chernoff Faces can be seen by viewing the Requirements model (accessible via the Models/Requirements menu) or the Risk model (accessible via the Models/Risk menu).

16.6.1 Template Requirements

CAIRIS Home System ▾ Requirements ▾ Risk ▾ UX ▾ Models ▾ Options ▾ Search Search sfaily ▾

Home / Template requirements / Policy rules

Name

Policy rules

Type

Security ▾

Asset

Rule ▾

Specification

The Organisational Security Policy shall be modelled as a set of access rules.

Fit Criterion

None

Rationale

Defines organisational policy for network access.

Update Cancel

alt TemplateRequirementDialog

You can specify libraries of template requirements that you might form the basis of security and architectural patterns.

These can be added, updated, and deleted in much the same way as other CAIRIS objects.

16.7 Attack tree modelling with obstacles

Attack trees are a formal methodological way of describing the security of systems. Together with Data Flow Diagrams (DFDs) these are a standard for visualising threat models.

Because obstacle models are represented using the same top-down notation as attack trees, they are a good candidate for representing attack.

You can import attack trees represented as Dot files directly into a CAIRIS model. See the **‘Importing and Exporting models’** section for more details on how to import models into CAIRIS.

Use cases are sequences of actions a system performs that yields an observable result of value to a particular *actor*; in CAIRIS, *actors* are analogous to *roles*.

17.1 Adding, updating, or deleting a use cases

- Click on the Requirements/Use Case menu to open the Use Cases table, and click on the Add button to open the Use Case form.
- Enter a use case name, a short code, details of the author, and – in the Description folder – the objective of carrying out the use case. In the Actors folder, you should also add one or more roles that constitute the actors for this use case.
- Click on the Add button in the environment card, and select an environment to situate the use case in. This will add the new environment to the environment list.
- In the Preconditions folder, enter any pre-conditions that need to hold in this context of use before the use case begins.
- In the Postconditions folder, enter any post-conditions that need to hold in this context of use once the use case completes.
- In the Flow folder, click on the Add button in the Step table to add a step to the use case. In the Use Case Step dialog, you should describe how the actor or system interact within this step. You should then click on Update to add the step to the use case.
- Click on the Create button to add the new use case.
- Existing use cases can be modified by clicking on the use case in the Use Cases table, making the necessary changes, and clicking on the Update button.
- To delete a use case, select the use case to delete in the Use Cases, and click the Delete button. If any artifacts are dependent on this use case then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the use case dependencies and the use case itself, or No to cancel the deletion.

CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

Search

sfaily ▾

Home / Use cases / Modify Telemetry Software

Summary

Contribution

Use Case

Short Code

Author

Modify Telemetry Software

MTS

Anon

+

Actor

-

Instrument Technician

Objective

Modify telemetry software

Tags

+ Environment

Day

Night

Preconditions

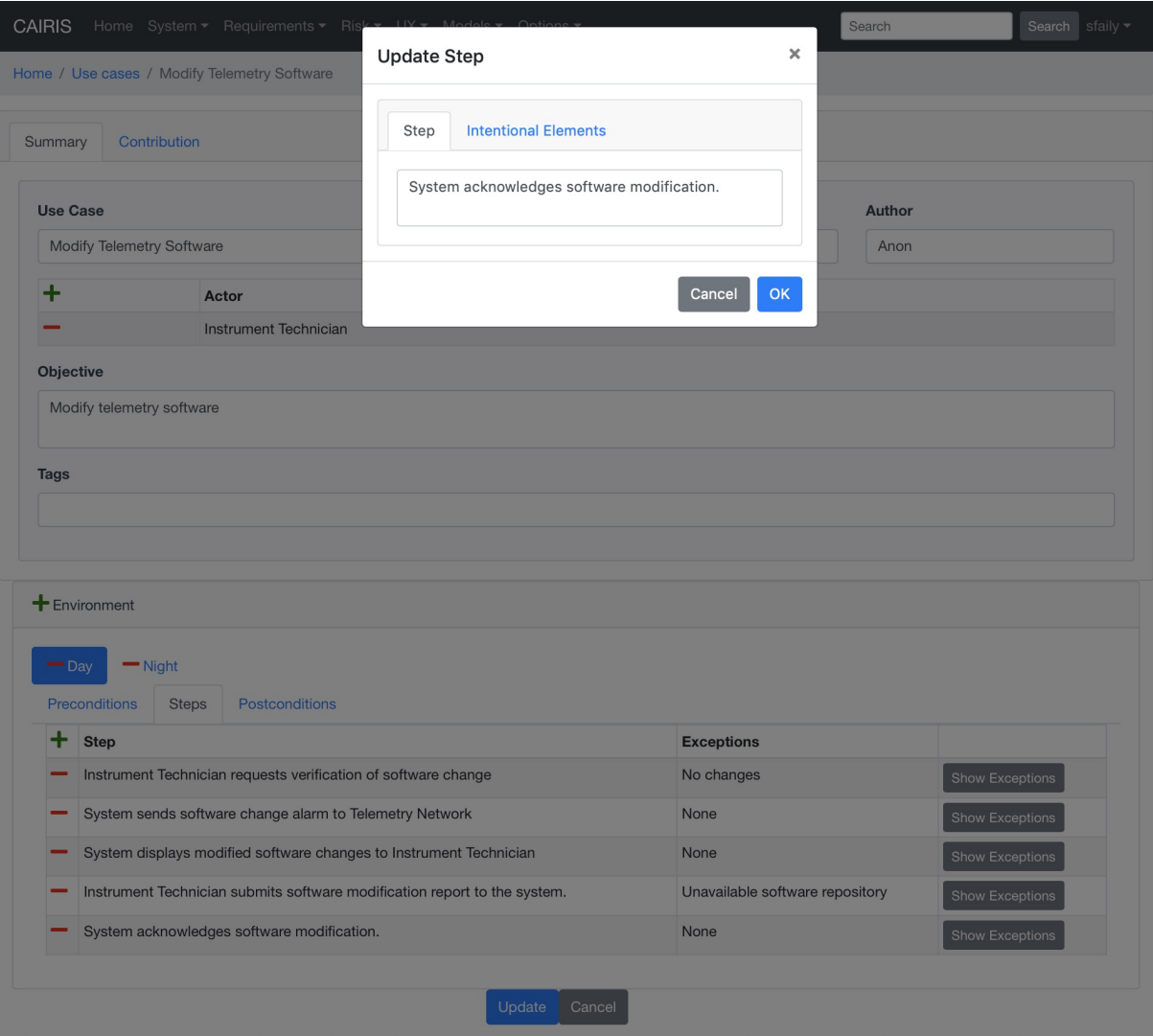
Steps

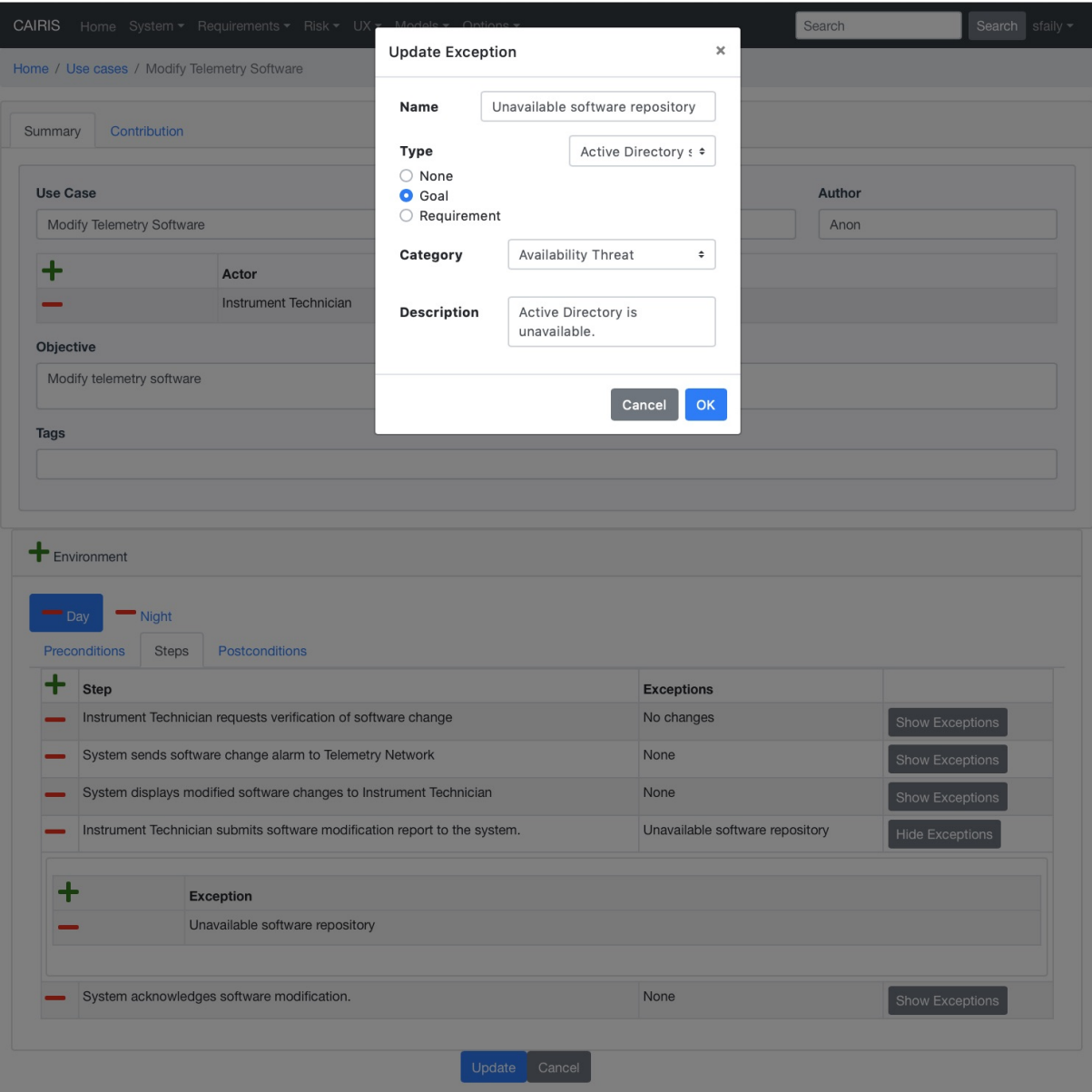
Postconditions

Software repository online. Alarm mechanisms online. Instrument technician authenticated with outstation. Instrument technician authenticated with telemetry network. Modified software on laptop and outstation.

Update

Cancel





17.2 Add exceptions to use case steps

- Select the step and click on the Add button in the Exceptions table.
- In the Use Case Step Exception dialog, enter the name of the exception, the category of threat, vulnerability, or usability conflict associated with this exception, and a definition of the exception.
- Select the goal or requirement that this step conflicts with, otherwise select the None radio button. Goals are visible only if you have added a sub-goal refinement relationship between goals and this use case. Requirements are visible only if you have added a manual 'Supported by' traceability link between requirements and this use case.
- Click Add to add the exception to the Exceptions table. When the use case is created or update, obstacles are generated based on exceptions associated with goals or requirements.
- Existing exceptions can be modified by double clicking on the step in the Exceptions table, making the necessary changes, and clicking on the Update button.

User goals and user goal models

CAIRIS supports the specification, modelling, and validation of user goal models. These models are based on a subset of the [Goal-oriented Requirements Language \(GRL\)](#) : a language for modelling intentional relationships between goals.

There are several reasons why you might find working with user goals useful.

- Expressing persona data using user goals can help elicit intentional relationships that support or refute aspects of a persona’s behaviour.
- Agent-oriented goal modelling language are popular in Requirements Engineering, making a user goal model a potential vehicle for interchange between RE methods, techniques, and tools.
- By exploring the way that user goals contribute to other user goals, it is possible to identify new requirements, threats, or vulnerabilities resulting from goals that are satisfied or denied.

User goals represent the intentional desires of actors, where actors are personas. Three types of user goals can be specified in CAIRIS:

- [Hard] goals are user goals that can be measurably satisfied.
- Soft goals are user goals with less well-defined success criteria that can be satisfied.
- Beliefs capture perceptions or opinions that are important to the actor.

User goal models can be generated in CAIRIS or, alternatively, can be exported to [jUCMNav](#).

18.1 Adding, updating, and deleting user goals

Before you can create a user goal, you first need to create a document reference. If document references represent the factoids upon which a persona is based, a user goal is this factoid expressed in intentional terms.

- To create a user goal, click on the UX/User Goals menu to open the user goals table, and click on the Add button to open the user goal form.
- Enter the name of the user goal. Because they expressed intentions, user goals should follow the naming convention of “goal [be] achieved”, e.g. “AJS task captured”.
- Select the persona associated with this user goal.
- Indicate whether the user goal is a belief, [hard] goal, or soft goal.

CAIRIS Home System Requirements Risk UX Models Options Search sfaily@bournemouth.ac.uk

Home / User goals / AJS tasks captured

User Goal

AJS tasks captured

Persona

Rick

Element Type

☐ Belief ☒ Goal ☐ Soft Goal

Reference Type

☒ Document ☐ Persona

Reference

Tasks entered into AJS are captured by WMS

Tasks are entered into Asset Job System are captured by Work Management System.

Initial Satisfaction

None

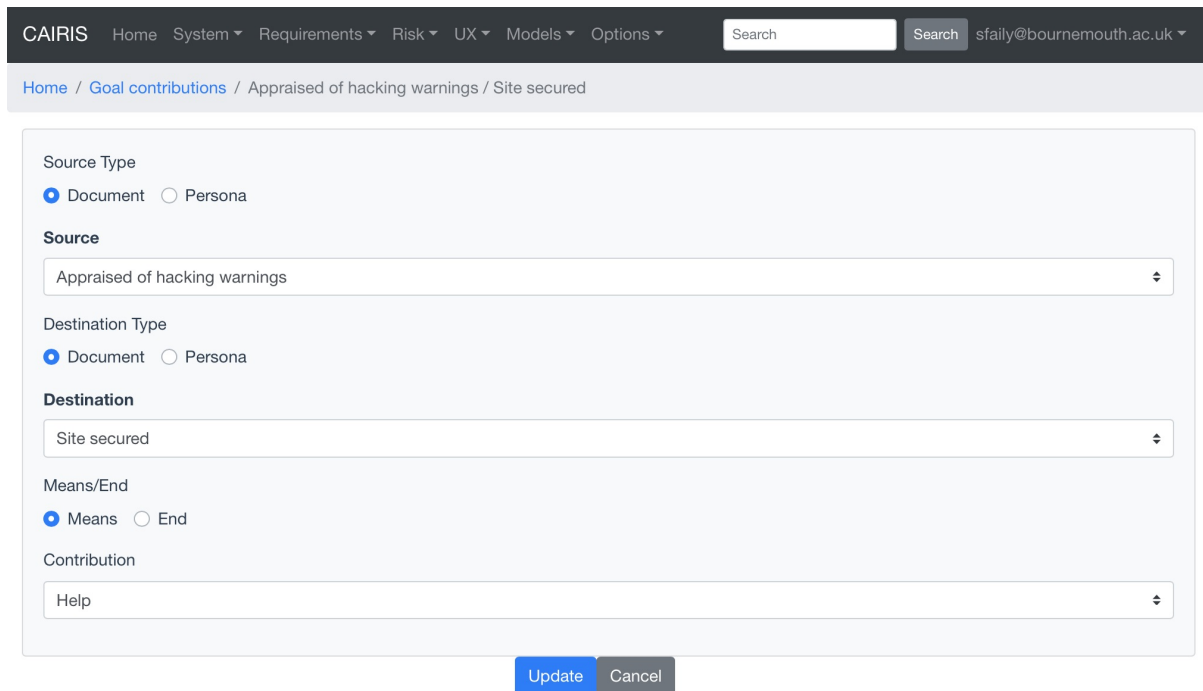
+ System Goal

Update Cancel

- Select the Reference grounding the user goal. If this is a document reference, select *Document* as the element type and select the document reference name from the combo box. If the user goal is based on a persona characteristic, select *Persona* as the element type. To help you phrase the user goal, details of the document reference or persona characteristic are displayed.
- If you wish, you can override the calculated satisfaction score with an initial satisfaction value. The available values are Satisfied (100), Weakly Satisfied (50), None (0), Weakly Denied (-50), Denied (50).
- If you wish to associate the user goal with a KAOS goal, click on the Add button in the System Goal table to select the goal.
- Click on the Create button to add a new user goal.
- Existing user goals can be modified by clicking on the user goal in the User Goals table, making the necessary changes, and clicking on the Update button.
- To delete a user goal, select the user goal to delete in the User Goals table, and click on the Delete button.

18.2 Adding, updating, and deleting user goal contributions

- To create a user goal contribution, click on the UX/User Goal Contributions menu to open the user goal contributions table, and click on the Add button to open the user goal contribution form.
- Depending on the reference grounding the source user goal, select *Document* or *Persona* as the source type, and select the source user goal name.
- Depending on the reference grounding the destination user goal, select *Document* or *Persona* as the source type, and select the destination user goal name.
- Indicate whether the source user goal is the *means* or the *end* of the user goal contribution link.



- Select the strength of the contribution link. The options available are Make (100), SomePositive (50), Help (25), Hurt (-25), SomeNegative (-50), and Break (-100).
- Click on the Create button to add a new user goal contribution.
- Existing contribution links can be modified by clicking on the contribution in the user goal contributions table, making the necessary changes, and clicking on the Update button.
- To delete a user goal contribution, select the contribution link to delete in the user goal contributions table, and click on the Delete button.

18.3 Task contributions

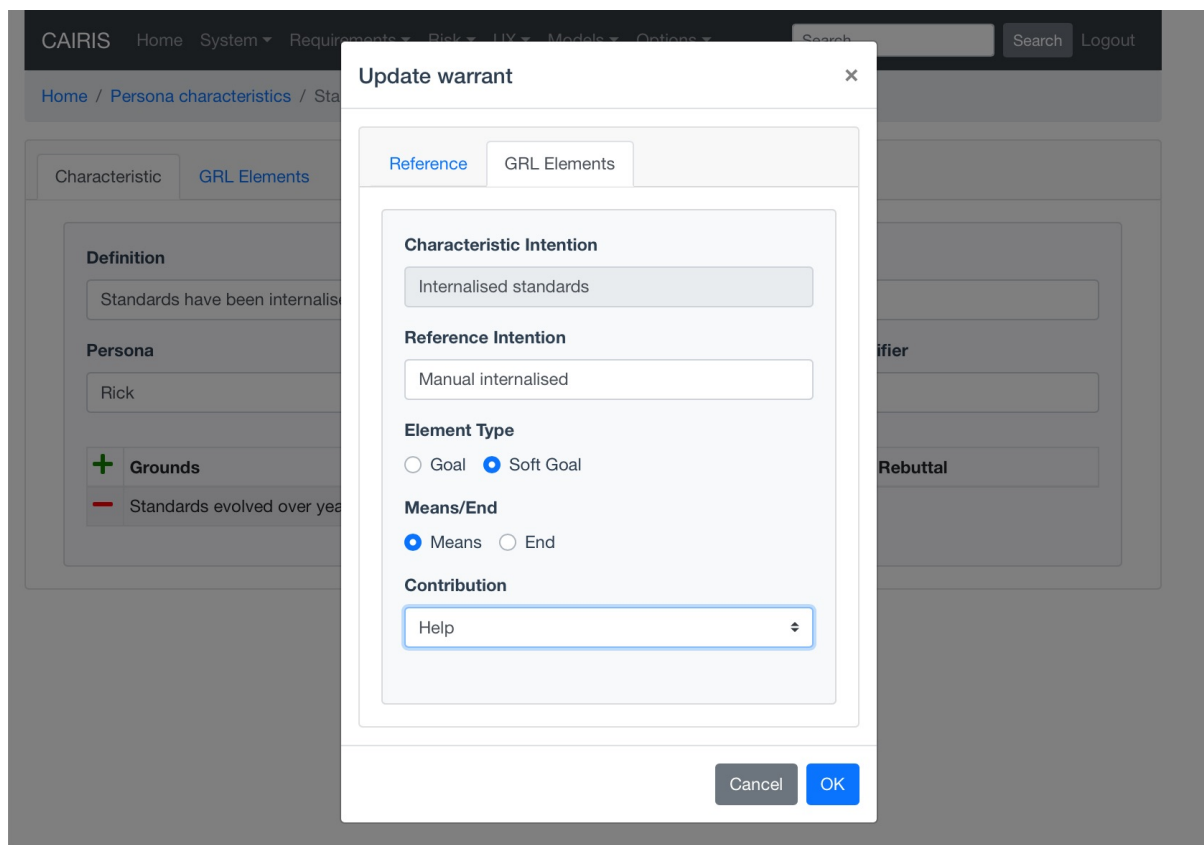
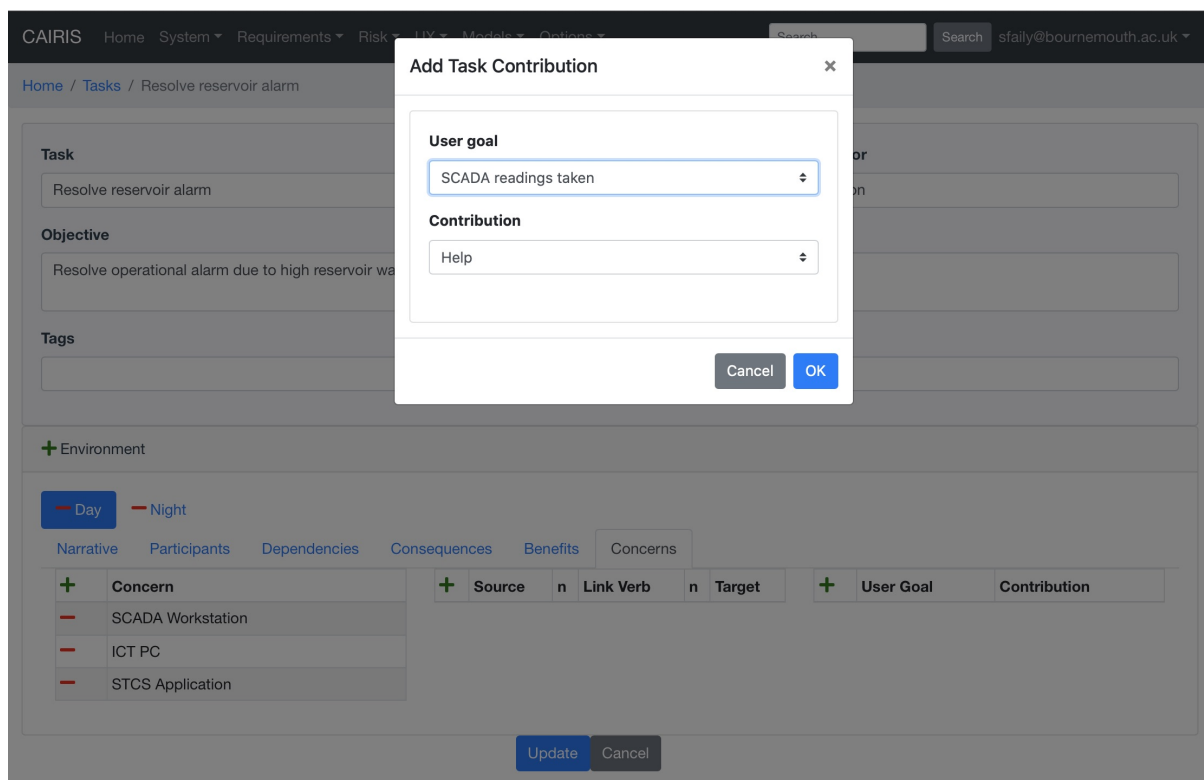
In addition to adding an initial satisfaction level for user goals, you can also set the contribution level that a task has on one or more user goals.

- To add such a contribution link, in the appropriate task, click on the Add button in the User Goal Contribution table in the Concerns folder for the appropriate task environment.
- From the Task Contribution dialog box, select the user goal concerned with the task in this environment and click Ok. The task-goal contribution link will be added when the task is created or updated.

18.4 Adding User goal elements to persona characteristics

User goals can be associated with persona characteristics, and their supporting grounds, warrants, or rebuttals. User goals drawn from persona characteristics are implicitly linked with user goals associated with these grounds/warrants/rebuttal elements, so adding user goals while working with persona characteristics is a good way of initially specifying user goal models.

- To add these User goal elements, open the persona characteristic you want to update, and click on the User Goal Elements folder.
- Select the Element type for the user goal. This can be either a belief, goal, soft goal, or task (tasks are relevant only if exporting to jUCMNav).



- Enter a user goal that expresses the characteristic in intentional terms.

CAIRIS Home System Requirements Risk UX Models Options Search sfaily@bournemouth.ac.uk

Home / Persona characteristics / Area of responsibility is large and unpredictable

Characteristic User Goal Elements

Element Type

☐ Belief ☒ Goal ☐ Soft Goal ☐ Task

User Goal

Large area managed

Update Cancel

- For each appropriate grounds, warrant, and rebuttal reference, click on the reference to open the characteristic reference dialog.
- Expresses the ground, warrant, or rebuttal reference in intentional terms.
- Select the element type for this synopsis this can be a belief, goal or soft goal.
- Given the intentional relationship between this element and the belief, goal, softgoal, or task associated with the persona characteristic, indicate whether this element is a means for achieving the characteristic element's end by selecting *Means* in the Means/End combo box. Alternatively, if the characteristic's element is a means for achieving this user goal element end then select *End*.
- Use the Contribution box to indicate how much this reference contributes to achieving its means or end. Possible values are Make (100), SomePositive (50), Help (25), Hurt (-25), SomeNegative (-50), and Break (-100).
- Click on the Save button to update the persona characteristic, and close the dialog.
- Click on the Update button on the persona characteristic form to save the persona characteristic.

18.5 Adding GRL elements to use cases (jUCMNav export only)

Use cases can make a contribution to GRL elements associated with persona characteristics. These use cases are associated with GRL goals, and the use case steps are refined as GRL tasks. These are associated with either asset, component, or role actors.

- To add these GRL elements and contribution relationships, open the use case to be updated, and select the Contribution folder.
- Select the goal or soft goal the use case contributes to, indicate whether the use case is a means or an end in the intention relationships, and – using the Contribution box – indicate how much the use case contributes to achieving its means or ends.
- Click on the Flow folder, and double click on the step you want to associate the GRL task with.
- Enter a synopsis that expresses the use case step in intentional terms.
- Select the GRL actor type and actor to associate the GRL task with. Permissible actor types are assets, components, and roles.
- Click on the Update button to update the use case step, and close the dialog.
- Click on the Update button on the use case form to save the use case.

CAIRIS

HomeSystemRequirementsRiskUXModelsOptions

Search

Search

Logout

Home / Use cases / Authenticate with ICT services

SummaryContribution

	Intention	Means/End	Contribution
+	Internalised standards	end	Help

Environment

DayNight

PreconditionsStepsPostconditions

Plant Operator not authenticated with ICT services

UpdateCancel

CAIRIS

HomeSystemRequirementsRiskUXModelsOptions

Search

Search

Logout

Home / Use cases / Authenticate with ICT services

SummaryContribution

	Intention
+	Internalised standards

Environment

DayNight

PreconditionsStepsPostconditions

Step

Plant operator enters credentials.

System verifies credentials.

UpdateStep

StepIntentional Elements

Intention

Enter credentials

Actor

Asset

Role

Component

Plant Operator

CancelOK

Step 1

Exception

UpdateCancel

18.6 Viewing a user goal model



To view the user goal model, click on the Models/User Goal model. Like other models, clicking on model nodes provides more details on the user goal or task.

18.7 Working with workbooks

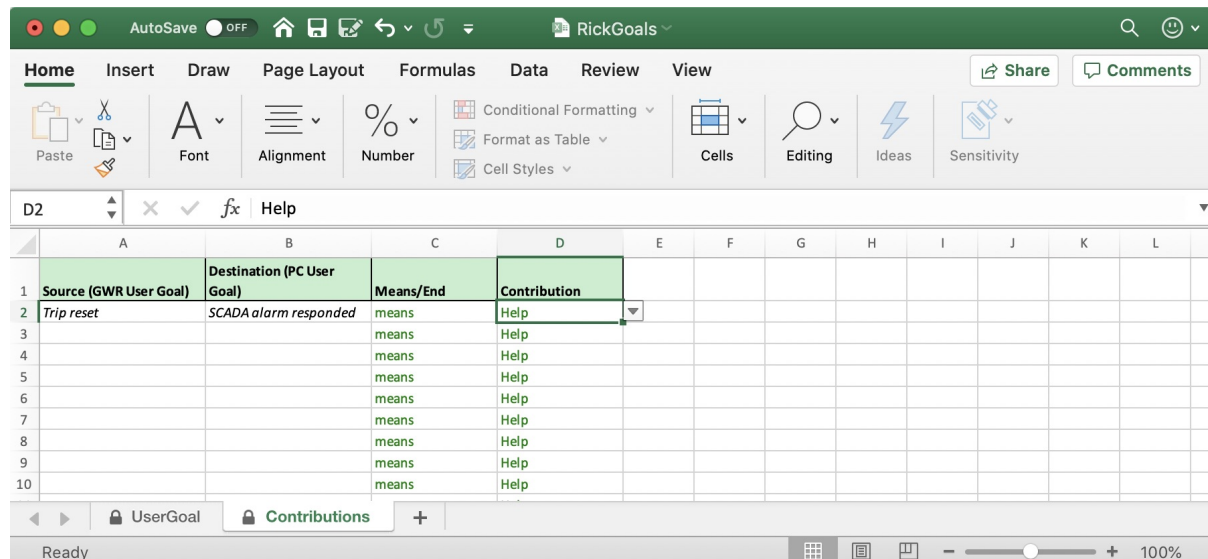
CAIRIS can generate an Excel workbook for capturing user goals and contribution links from persona characteristics. To create a workbook, select the System/Export menu, click on the *User goals (Workbook)* radio button, enter the spreadsheet file to be created, and click on the Export button.

Note: If you have server access, you can also run the `cairis/bin/ug2wb.py` script, indicating the user account, database, and name of the XLSX file to be generated, i.e. `./ug2wb.py --user test --database default RickGoals.xlsx`.

The generated Excel workbook (which is compatible with LibreOffice), contains UserGoal and UserContribution worksheets. Edited cells for both sheets are coloured green.

Reference	Description	Persona	persona/document_reference	Element Type	User Goal	Initial Satisfaction
1	React to alarms raised by SCADA	Rick	persona	goal	SCADA alarm responded	None
2	Readings periodically taken from SCADA	Rick	persona	goal		None
3	Managers need to authorise what they can and cannot do	Rick	persona	goal		None
4	Area of responsibility is large and unpredictable	Rick	persona	goal		None
5	Routine varies by time of day	Rick	persona	goal		None
6	Scheduled tasks issued by AIS	Rick	persona	goal		None
7	Process decisions may be weather based.	Rick	persona	goal		None
8	Processes regularly checked against spec	Rick	persona	goal		None
9	Liases with TIS technicians and the Environmental Agency	Rick	persona	goal		None
10	Samples are taken throughout the work and recorded	Rick	persona	goal		None
11						

The UserGoal worksheet is pre-populated with read-only data on the persona characteristic or document reference name, its description, the persona it is associated with, and an indicator to whether the *reference* corresponds to a persona [characteristic] or document reference. When completing the worksheet, you should indicate the intentional elements associated with the persona characteristics or document references providing their grounds, warrants, or rebuttals. You should also indicate the element type (goal, softgoal, or belief), and - if you wish - the initial satisfaction level using the dropdown lists provided. When generating a CAIRIS model, new user goals will only be created if cell values for each row are complete.



The source and destination cells in the ContributionsSheet are pre-populated once user goals have been added in the UserGoal sheet, so you only need to ensure the means/end and contribution links are set. When generating a CAIRIS model, contribution links will only be created if both Source AND Destination values have been set, i.e. their associated user goals have been defined.

To re-import the completed workbook back to CAIRIS, select the System/Import menu, select *User goals (Workbook)* from the dropdown box, select the workbook to be imported, and click on the Import button.

Note: If you have server access, you can also run the `cairis/bin/wb2ug.py` script, indicating the name of the XLSX file to be imported and the name of the CAIRIS model file to be created, i.e. `./wb2ug.py --xlsx RickGoals.xlsx RickGoals.xml`. The resulting model can be imported into CAIRIS, but take care not to overwrite existing data.

18.8 Generating a jUCMNav compatible GRL model

- To generate a GRL model, select the System/Export GRL menu to open the Export GRL modal dialog.
- Select the Environment, Task, and Persona to create the GRL model for, together with the name of the output GRL file. Persona GRL elements will be present in the exported model only GRL elements have been associated with persona cases. Task GRL elements will be present only if (i) a traceability link has been added between the use cases where GRL elements are elements.
- Click on the Export button to generate a GRL file.
- Assuming you have a project open in jUCMNav, you can import this GRL file by selecting the File/Import menu in Eclipse to open the Import wizard, selecting the Other | Import UCM / GRL / URN option, and then selecting the generated GRL file.

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
Logout

Home / Export

Model
☐ Model ☒ GRL ☐ Architectural Pattern

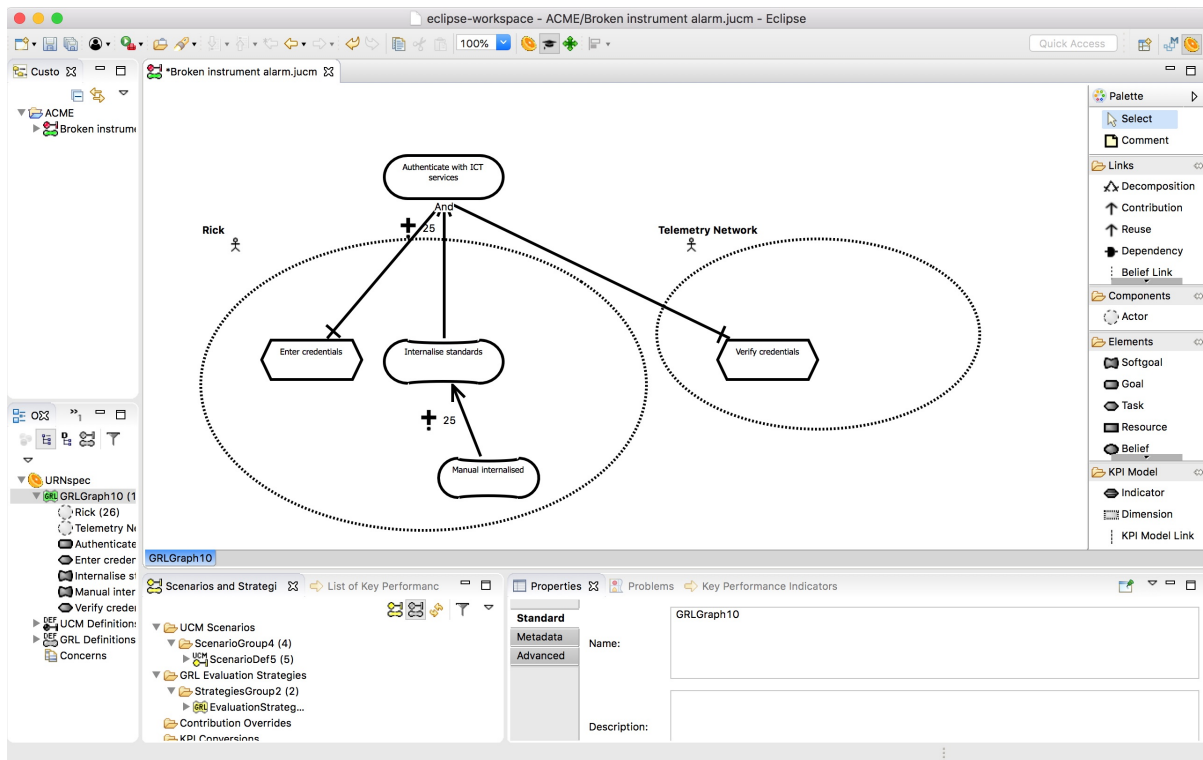
Environment
Day

Task
Broken instrument alarm

Persona
all

File name
test.grl

Export
Cancel



CAIRIS supports the modelling of strategic dependencies between roles, where these *dependers* depend on *dependees* for a dependum (goals, assets, or tasks).

19.1 Adding, updating, and deleting a dependency

The screenshot shows a web browser window with the URL `demo.cairis.org`. The CAIRIS navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a Logout button. The breadcrumb trail indicates the current path: Home / Dependency / Day / Information Security Manager / Vendor / Product security accreditation.

The main form contains the following fields:

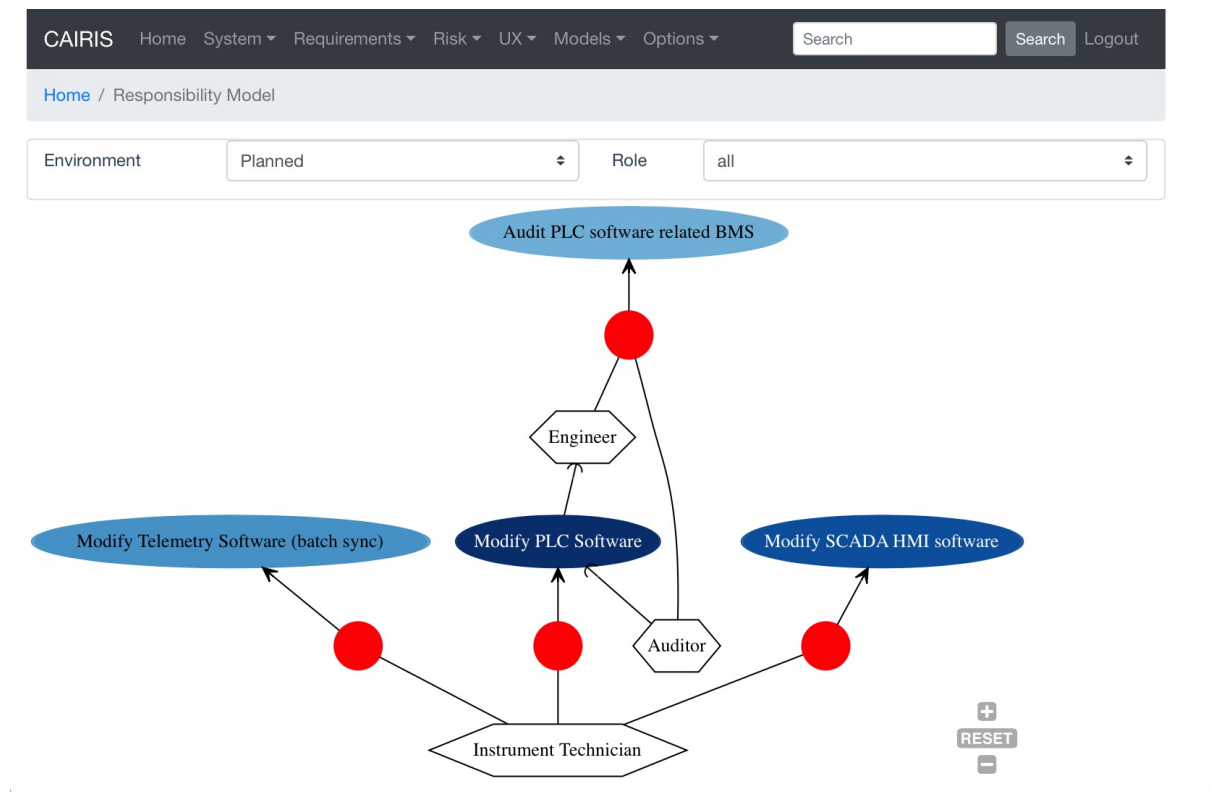
- Environment:** A dropdown menu with 'Day' selected.
- Depender:** A dropdown menu with 'Information Security Manager' selected.
- Dependency:** A section with three tabs: 'Asset', 'Goal', and 'Task'. The 'Task' tab is active, showing a dropdown menu with 'Product security accreditation' selected.
- Dependee:** A dropdown menu with 'Vendor' selected.
- Rationale:** A text input field containing the text: 'ACME rely on vendors to provide proof of a product's level of security.'

At the bottom of the form are two buttons: 'Update' (in blue) and 'Cancel' (in grey).

19.2 Viewing dependencies

Dependencies can be viewed by Responsibility models by clicking on the Models/Responsibility menu, and selecting the environment to view the environment for.

Dependencies are indicated by the rounded arrows that flow from the depender to the dependee through the dependum.



19.3 Introducing Personal data into CAIRIS using dependencies

Personal data can be recognised in CAIRIS by carrying out the following steps.

1. Create or edit a *Data Subject* role, and associate with this with a new persona; this could represent the natural person whose data is subject to processing, or a data controller or processor from another system contributing this data. The persona will be useful for capturing any assumptions or expectations this entity might have.
2. Create or edit a *Data Controller* role.
3. Create a dependency between the Data Subject dependee role and the Data Controller depender, where the dependum is a newly created or existing asset that you wish to designate as personal data.

Security Patterns are solution structures, which prescribe a solution to a security problem arising in a context. Many components and connectors in secure system architectures are instances of security patterns but, in many cases, the reasoning for a given pattern's inclusion is not always clear. The requirements needed to realise these patterns are also often omitted, making the job of reasoning about the consequences of situating the pattern difficult. Moreover, security patterns may be described in a context, but not all collaborating assets in a security pattern may be evident in all possible contexts of a system's use. The following sections describe how CAIRIS treats security patterns and deals with these weaknesses.

Security Patterns in CAIRIS consist of the following elements:

- A description of the context a pattern is relevant for.
- A problem statement motivating the need for the pattern.
- A solution statement describing the intrinsics of the pattern.
- The pattern structure, modelled as associations between collaborating asset classes.
- A set of requirements, which need to be fulfilled in order to realise the pattern. These requirements are based on template requirements.

Before a security pattern can be defined in CAIRIS, template assets – which represent the collaborating asset classes – need to be first defined.

Before a security pattern can be situated in CAIRIS environments, the environments themselves need to be first created.

20.1 Create a template asset

Template assets can be best described as context-free assets. When they are created, template assets do not form part of analysis unless they are implicitly introduced. This 'implicit introduction' occurs when a security pattern is situated.

The Template Patterns dialog can be opened by selecting the Options/Template Assets menu option.

The process for creating, updating, and deleting a template asset is almost identical to the processes used for normal assets. The only difference is the lack of environment-specific properties. Security properties are only defined once for the asset.

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
sfaily

Home / Template assets / Stateful Firewall

Summary
Interfaces

Asset

Stateful Firewall

Shortcode

SFF

Type

Systems

Description

Filters incoming and outgoing network traffic in a computer system based on state information derived from past communications.

Significance

None

Surface Type

Privileged application

Access Right

trusted

Tags

Update
Cancel

To situate an asset in an environment, click on the situate button in the template assets list, and specify the environments to situate the template asset in. After a template asset is situated within an environment, these properties should be revised in the assets generated on the basis of these. This is because the values associated with the template asset properties may not be inline with assumptions held about Low, Medium, and High assets in the specification being developed.

20.2 Create a security pattern

- Select the Risks/Security Patterns menu option to open the Security Patterns table, and click on the Add button to open the Security Pattern dialog form.
- Enter the security pattern name, and description of the context the security pattern is relevant for,
- a problem description motivating the security pattern, and the intrinsics of how the security pattern solves the pre-defined problem.
- Click on the Structure tab, and the Add button in the associations table to add associations between template assets; these associations form the collaborative structure for the pattern. The procedure for entering associations is based on that used for associating assets.
- Click on the Requirements tab, and the Add button in the requirements table to add names of template requirements needing to be satisfied to realise the pattern.
- Click on the Create button to add the new security pattern.
- Existing security patterns can be modified by double clicking on the security pattern in the security patterns table, making the necessary changes, and clicking on the Update button.
- To delete a security pattern, click on the Delete button besides pattern to delete in the security patterns table.

CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

Searchsfaily ▾

Home / Security patterns / Packet Filter Firewall

Summary

Structure

Requirements

Name

Packet Filter Firewall

Context

Computer systems on a local network connected to the Internet and to other networks with different levels of trust. A host in a local network receives and sends traffic to other networks. This traffic has several layers or levels. The most basic level is the IP level, made up of packets consisting of headers and bodies (payloads). The headers include the source and destination addresses as well as other routing information, while the bodies include the message payloads.

Problem

Some hosts on other networks may try to attack the local network through their IP-level payloads. We also need to communicate with other networks, so isolating our network is not an option. The protection mechanism should reflect the security policies of the organisation, and any protection mechanism should be transparent to the user.

Solution

A Packet Filter Firewall intercepts all traffic coming and going from a port P and inspects its packets. Those coming from or going to mistrusted addresses are rejected

Update

Cancel

20.3 Situate a security pattern

- To introduce a security pattern into the working project, select the Risks/Security Patterns menu, click on the situate button for the pattern to be situated.
- Select the environment to situate the environment in, and click on Ok to situate the pattern.

Template assets will be instantiated as assets, and situate in the stipulated assets. Requirements associated with the pattern, will be introduce and associated with the stipulated assets in the pattern definition. These assets will be ordered based on the order of definition in the pattern structure.

Vulnerabilities are weaknesses of a system that are liable to exploitation.

21.1 Create a vulnerability

CAIRIS Home System ▾ Requirements ▾ Risk ▾ UX ▾ Models ▾ Options ▾ sfaily ▾

Home / Vulnerabilities / Certificate ubiquity

Vulnerability

Certificate ubiquity

Type

Configuration ▾

Description

Certificates can be shared, allowing unauthorised users access to the NeuroGrid portal.

Tags

+ Environment

Psychosis

Stroke

Severity

Critical ▾

+ Asset

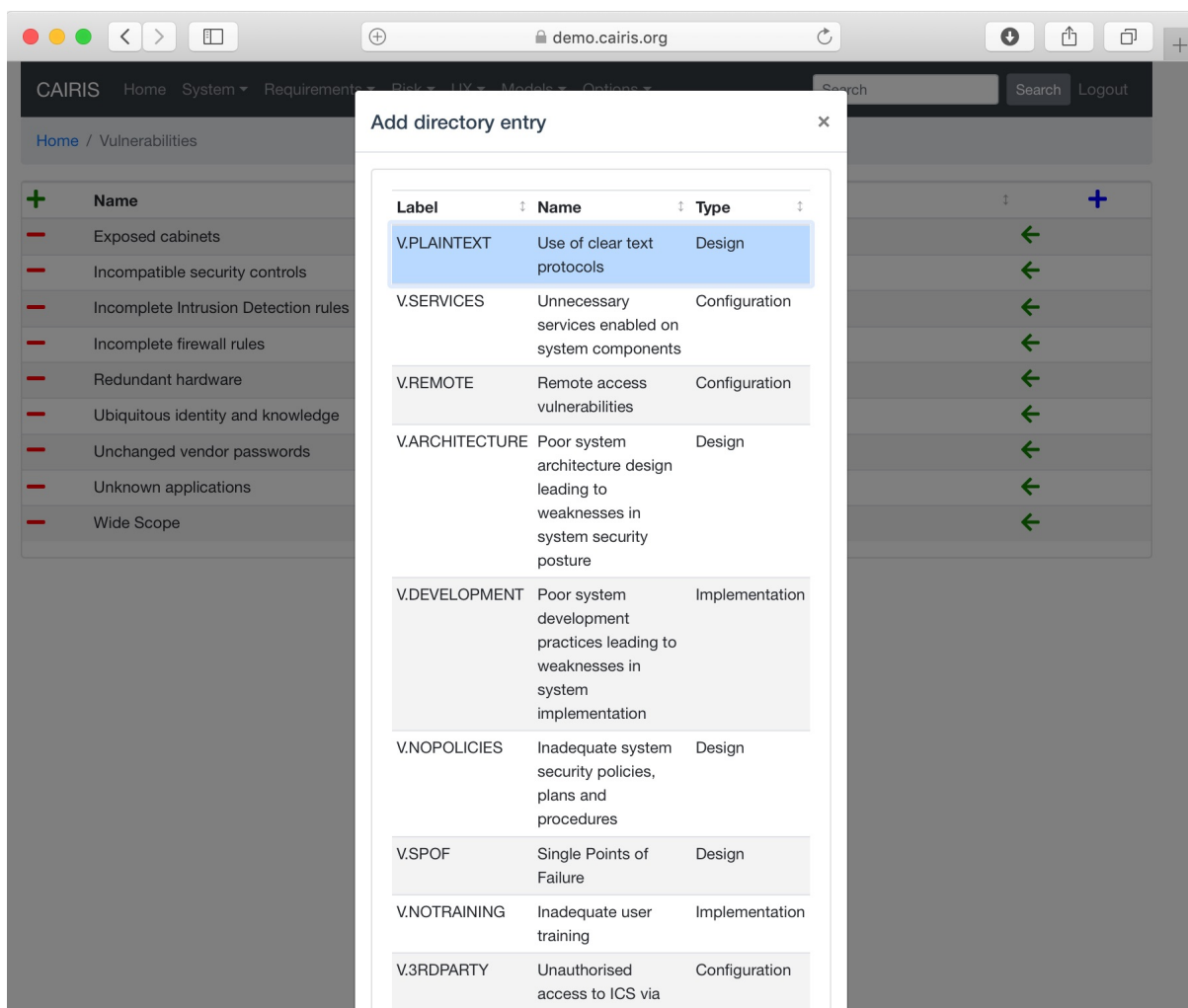
User certificate

Update

Cancel

- Select the Risk/Vulnerabilities menu to open the Vulnerabilities table.
- Click on the Add button to open the Vulnerability form.
- Enter the vulnerability name and description, and select the vulnerability type from the combo box.
- Click on the Add button in the environment card, and select an environment to situate the vulnerability in. This will add the new environment to the environment list.
- Select the vulnerability's severity for this environment, and add exposed assets by clicking on the Add button in the assets table, and selecting one or more assets from the selected environment.
- Click on the Create button to add the new vulnerability.
- Existing vulnerabilities can be modified by clicking on the vulnerability in the Vulnerabilities table box, making the necessary changes, and clicking on the Update button.
- To delete an vulnerability, click on the delete button next to the vulnerability to be deleted in the Vulnerabilities table. If any artifacts are dependent on this vulnerability then a dialog box stating these dependencies are displayed. The user has the option of electing Yes to remove the vulnerability dependencies and the vulnerability itself, or No to cancel the deletion.

21.2 Introducing template threats and vulnerabilities



Libraries of template vulnerabilities can be imported into the CAIRIS database and introduced to the current CAIRIS model. Examples of such libraries in `cairis/examples/directories`. To import one of these, click on the blue Add button at from the top of Vulnerabilities or Threats table to open the Introduce from vulnerability directory

dialog. When a vulnerability is selected, the Vulnerability form is opened, and pre-populated with information from the directory entry.

The screenshot shows a web browser window with the URL `demo.cairis.org`. The CAIRIS navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a Logout button. The breadcrumb trail indicates the current location: Home / Vulnerabilities / V.PLAINTEXT.

The main form is titled "Vulnerability" and contains the following fields:

- Vulnerability:** A text input field containing "V.PLAINTEXT".
- Type:** A dropdown menu with "Design" selected.
- Description:** A text area containing the text: "Use of clear text protocols: The use of clear text protocols and the transmission of business and control data unencrypted over insecure communication channels (e.g. FTP, TELNET). Reference: SPP-ICSv1.0".
- Tags:** An empty text input field.
- + Environment:** A section header with a plus icon, followed by an empty text input field.

At the bottom of the form are two buttons: "Create" (blue) and "Cancel" (grey).

Attackers launch attacks in the form of threats. Attackers are similar to personas in that they fulfill one or more roles, and can be personalised with additional information.

Certain capabilities and motivations may be associated with attackers. CAIRIS is pre-loaded with a selection of these, but these can be modified, or new capabilities and motivations created by selecting the Options/Capabilities or Options/Motivations menu options.


22.1 Adding, updating, and deleting an attacker

- Select the Risk/Attackers toolbar menu to open the Attackers table, and click on the Add button to open the Attacker form.
- Enter the attacker name, and a description for the attacker.
- If you have decided to personalise the attacker with a picture, this can be added by clicking on avatar silhouette next to the attacker description, and selecting a image to represent the attacker. Permitted image types are jpg, png, giff, and bmp.
- Click on the Add button in the environment card, and select an environment to situate the attacker in. This will add the new environment to the environment list.
- Click on the Add button on the Roles table to associate one or more roles to the attacker.
- Click on the Add button on the Motivation and Capability tables to add one or more motive and capability values. For the capability, a value of Low, Medium, or High also needs to be selected.
- Click on the Create button to add the new attacker.
- Existing attackers can be modified by clicking on the attacker in the Attackers table, making the necessary changes, and clicking on the Update button.
- To delete an attacker, click on the Delete button next to the attacker to be removed in the Attackers table. If any artifacts are dependent on this attacker then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the attacker dependencies and the attacker itself, or No to cancel the deletion.

demo.cairis.org

CAIRIS Home System Requirements Risk UX Models Options Search Search Logout

Home / Attackers / Carol



Attacker

Tags

Description

Carol is a freelance journalist working in the South East of England. Having heard stories about data theft, she is currently investigating a number of e-Science projects, including NeuroGrid, to see if she can find a story.

As security is a topical subject in the media, Carol has been able to secure some seed money from a national newspaper to fund her investigation.

+ Environment

Psychosis

Stroke

+ Role
Social Engineer

+ Motivation
Headlines/press

+ Capability	Value
Resources/Personnel and Time	Medium
Resources/Funding	Low

Update

Cancel

23.1 Adding, updating, and deleting a threat

Threats are synonymous with attacks, and can therefore only be defined if an associated attacker has also been defined. Like vulnerabilities, threats are associated with one or more assets. However, threats may also target certain security properties as well, in line with security values that an attacker wishes to exploit.

A threat is also of a certain type. CAIRIS is pre-loaded with a selection of these, but these can be modified, or new threat types created by selecting the Options/Threat Types menu option.

- Select the Risks/Threats menu to open the Threats table, and click on the Add button to open the Threat form.
- Enter the threat name, the method taken by an attacker to release the threat, and select the threat type.
- Click on the Add button in the environment card, and select an environment to situate the threat in. This will add the new environment to the environment list.
- Select the threat's likelihood for this environment
- Associate attackers with this threat by clicking on the Add button above the Attacker table, and selecting one or more attackers specific to the environment.
- Add threatened assets by clicking on the Add button above the Assets table, and selecting one or more assets specific to the environment.
- Add the security properties to this threat by clicking on the Add button above the properties table, and selecting a security property, value, and rationale.
- Click on the Create button to add the new threat.
- Existing threats can be modified by clicking on the threat in the Threats table, making the necessary changes, and clicking on the Update button.
- To delete a threat, click on the Delete button threat next to the threat to be removed in the Threats table. If any artifacts are dependent on this attacker then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the threat dependencies and the threat itself, or No to cancel the deletion.

CAIRIS

HomeSystem▼Requirements▼Risk▼UX▼Models▼Options▼

Search

Search

sfaily▼

Home / Threats / False sensor readings

Threat

False sensor readings

Type

Electronic/Phishing and Spoofing

Method

False readings are sent to SCADA workstations

Tags

+ Environment

Day

Likelihood

Improbable

+ Attacker

Victor

+ Asset

SCADA Workstation

EnterpriseSCADA Server

+ Property

Integrity

Availability

Value

Medium

High

Rationale

Victor wants to corrupt the broader SCADA infrastructure.

Victor wants to stop water treatment.

Update

Cancel

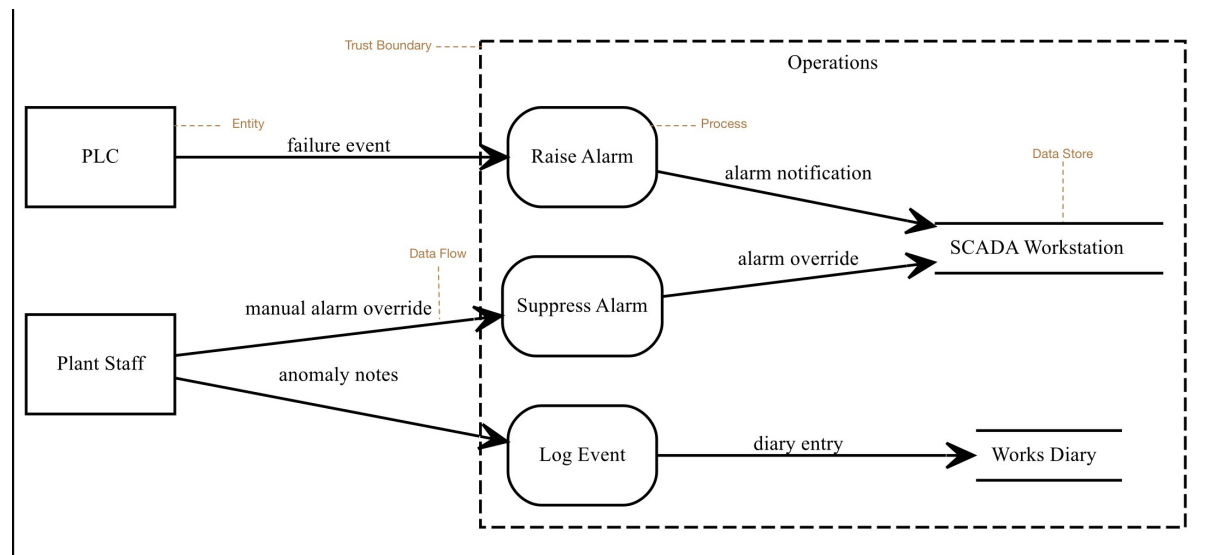
98

Chapter 23. Threats

CAIRIS supports two different techniques for threat modelling.

24.1 Data flows and Data Flow Diagrams

Data flow diagrams (DFDs) are graphical models that model the flow of information (data flows) between external human or system actors external to the system (entities), activities that manipulate data (processes), and persistent data storage (data stores). Together with attack trees, in threat modelling, DFD model elements can be encompassed by *trust boundaries*; these occur where entities with different privileges interact.



24.1.1 Adding, updating, and deleting entities, processes, and data stores

Entities are synonyms for assets of type *Systems*, *Hardware*, or *People*. Data stores are synonyms for assets of type *Information*. To add, update, or delete entities and data stores, you need to add, delete or update the synonymous asset.

Processes are synonyms for use cases. To add, update, or delete processes, you need to add, delete or update the synonymous use cases.

24.1.2 Adding, updating, or deleting data flows

The screenshot shows the 'Dataflow' dialog in the CAIRIS application. The dialog is titled 'Dataflow' and contains the following elements:

- Dataflow:** A text input field containing 'failure event'.
- Environment:** A dropdown menu showing 'Day'.
- Type:** A dropdown menu showing 'Information'.
- From:** A dropdown menu showing 'PLC'. Below it are three radio buttons: 'Entity' (selected), 'Datastore', and 'Process'.
- To:** A dropdown menu showing 'Raise Alarm'. Below it are three radio buttons: 'Entity' (selected), 'Datastore', and 'Process'.
- Asset Table:** A table with two columns: a plus icon and 'Asset'. It contains one row with a minus icon and 'PLC Event'.
- Obstacle Table:** A table with two columns: a plus icon and 'Obstacle'. It is currently empty.
- Tags:** A text input field with the placeholder 'Enter new tags separated by comma'.
- Buttons:** 'Update' and 'Cancel' buttons at the bottom right.

- To add a data flow, select the UX / Data Flows menu to open the Data Flows table. Click on the Add button to open a dialog for adding a new data flow.
- Enter the name for the data flow, select the environment the data flow is specific to, and select the data flow type. You should also select the *from* and *to* types associated with the flow. These types are Entities, Data Stores, and Processes, where Entities are information, hardware, or people assets, Data Stores are information assets, and Processes are use cases.
- Click the Add button in the Asset table to choose one or more assets carried by this data flow.
- Should there be any obstructions to the data flow, click the Add button in the Obstacle table to add associated obstacles.
- Click on the Create button to add the data flow to the Data Flows table.
- An existing data flow can be edited by clicking on a data flow in the Data Flow table, updating any aspect of the data flow, and clicking on the Update button.
- Data flows can be deleted by clicking on the Delete button associated with the data flow to be removed in the Data Flows table.

24.1.3 Adding, updating, or deleting trust boundaries

alt Trust Boundary dialog

- To add a trust boundary, select the UX / Trust Boundaries menu to open the Trust Boundaries table. Click on the Add button to open a dialog for adding a new trust boundary.
- Enter the name, select the type, and enter a description for the trust boundary.

CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

Search

User ▾

[Home](#) / [Trust boundaries](#) / Operations

Trust Boundary

Operations

Type

General ▾

Description

All processes and objects associated with plant operations is considered trustworthy.

Tags

Enter new tags separated by comma

+ Environment

Day

Privilege

None ▾

+ -	Component	Type
-	Raise Alarm	process
-	Log Event	process
-	Suppress Alarm	process
-	Works Diary	datastore
-	SCADA Workstation	datastore

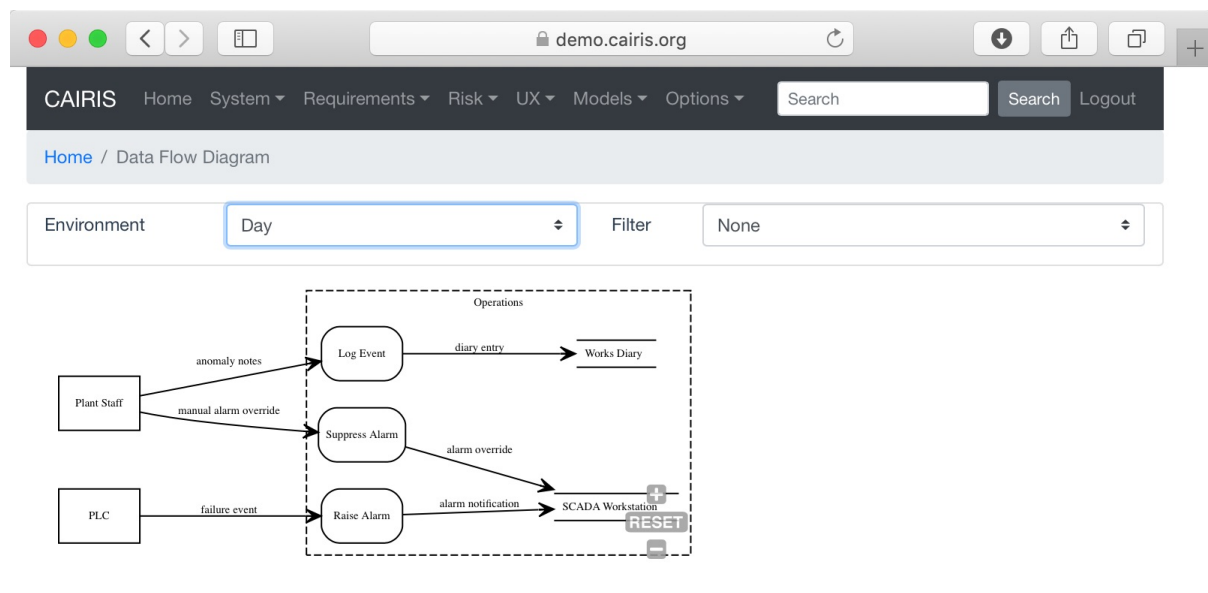
Update

Cancel

- Click on the Add button in the environment card, and select an environment to situate the trust boundary in. This will add the new environment to the environment list.
- Click the Add button in the Components table to situate a process or data store within this environment specific trust boundary.
- Select the level of privilege that the components in this trust boundary operate at.
- Click on the Create button to add the trust boundary to the Trust Boundary table.
- An existing trust boundary can be edited by clicking on a trust boundary in the Trust Boundaries table, updating any aspect of the trust boundary, and clicking on the Update button.
- Data flows can be deleted by clicking on the Delete button associated with the trust boundary to be removed in the Trust Boundaries table.

24.1.4 Viewing Data Flow Diagrams

DFDs can be viewed by selecting the Models/Data Flow menu, and selecting the environment to view the model for.



alt DFD

By changing the environment name in the environment combo box, the DFD for a different environment can be viewed. The model can also be filtered by DFD model element.

By clicking on a model element, information about that artifact can be viewed.

For details on how to print DFDs as SVG files, see [Generating Documentation](#).

24.1.5 Modelling DFDs with diagrams.net

You can use [diagrams.net](#) to import DFDs into CAIRIS by following the steps below:

1. Create a new blank diagram in [diagrams.net](#).
2. Setup the CAIRIS DFD shape library by going to the File >> Open Library from >> URL menu, and entering the URL https://cairis.org/stencils/cairis_dfd.xml.
3. To add an entity, click on the square in the `cairis_dfd.xml` palette to place an entity on the canvas. Double click on the shape to set its label, which represents the entity name. When importing the model, if an asset corresponding with the entity does not exist, CAIRIS will create a corresponding asset with some default values.

4. To add a process, click on the rounded box in the `cairis_dfd.xml` palette to place a process on the canvas. Double click on the shape to set its label, which represents the process name. When importing the model, if a use case corresponding with the process does not exist, CAIRIS will create a corresponding use case (and associated role) with some default values.
5. To add a data store, click on the parallel lines in the `cairis_dfd.xml` palette to place a data store on the canvas. Double click on the shape to set its label, which represents the data store name. When importing the model, if an asset corresponding with the data store does not exist, CAIRIS will create a corresponding asset with some default values.
6. To add a data flow between DFD elements, click on the arrow in the `cairis_dfd.xml` palette to place a data flow on the canvas. Double click on the data flow to set its label, which represents the data flow name. Right click on the data flow and select Edit Data to set the assets carried in the flow. By default, this is set to *UndefinedInformation*. This should be changed to represent the information assets carried by the data flow. Multiple assets should be separated by a comma. When importing the model, if assets corresponding with this comma separated list do not exist, CAIRIS will create them.
7. To encompass processes and data stores in a trust boundary, click on the dashed square in the `cairis_dfd.xml` palette to place a trust boundary on the canvas. Right click on the shape and select Edit Data to set the trust boundary name. Once set, move the processes and data stores within the trust boundary. Please note that, as external systems, entities should not be placed within trust boundaries.
6. Once the diagram is ready, select the File >> Export as >> XML... menu option, unclick the Compressed tick box, click on the Export button, and enter the name of the diagram to be exported.
7. In CAIRIS, select the System >> Import menu to open the Import form. Select *diagrams.net (Data Flow Diagram)* from the Model combo box, click on the File button to choose the exported diagrams.net model to import, and select the environment to import the DFD into.

Note: We recommend you use the `cairis_dfd.xml` shape library when data flow diagramming, but you could - in theory - use any shape in diagrams.net to model DFD elements. However, you must ensure that you use the Edit Data option to add a `type` property to the shape, which should be set to a valid DFD type (entity, process, datastore, or trustboundary). You also need to set a name property for trust boundaries. Similarly, you also use any line to link DFD elements, but you need to use the Edit Data option to add a `assets` property and define at least one asset as its value.

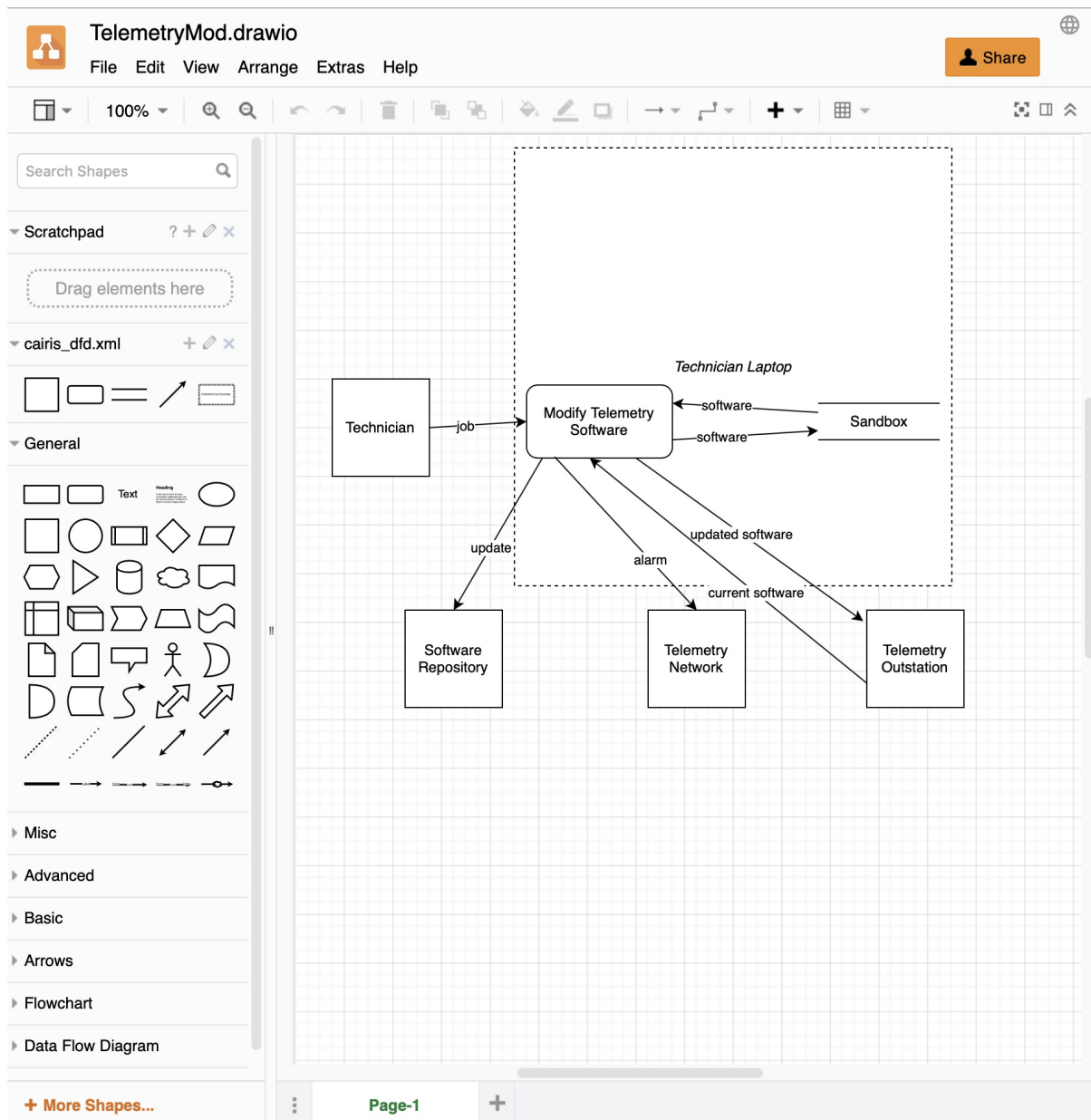
24.2 Attack trees

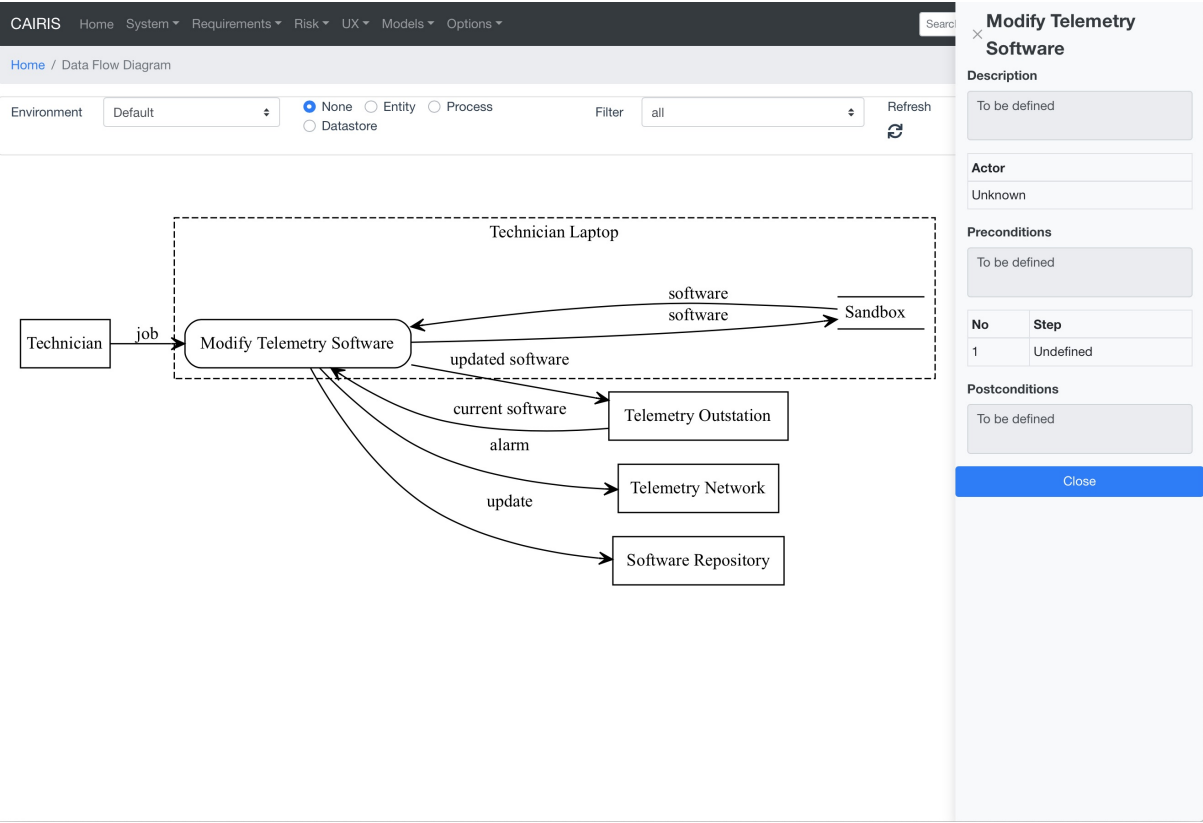
Attack trees are a formal, methodical way of describing the security of systems. They are a lightweight approach for modelling attacks; this is a good thing as they are simple enough that people can quickly create and contribute to them.

CAIRIS doesn't support attack trees, but obstacle models are represented using the same top-down approach notation as attack tree. This makes them a good candidate for representing the attacks, and the sort of things that need to hold for an attack to be successful.

Attack trees represented in [Dot](#) can be imported into CAIRIS by selecting the File/Import Model menu, selecting 'Attack Tree (Dot)' from the combo box, and choosing the .dot file to import. You will then be prompted for an environment to import the newly generated obstacles and obstacle associations into, together with the name of the contributor who created or imported the tree.

More details on using attack trees with CAIRIS can be found in this [blog post](#).





Using CAIRIS as tool-support for STPA

25.1 Overview

STPA (System-Theoretic Process Analysis) is a hazard analysis technique; it assumes accidents may be caused by unsafe interactions between system components, which may or may not have failed. CAIRIS can support the use of STPA because the concepts it supports are analogous with those required by STPA.

You may wish to use pen and paper to start your design exploration with STPA; this is entirely appropriate. However, as the STPA outputs become more complex, software tool support becomes useful. CAIRIS can help by providing automatic traceability between STPA elements, automatic generation of visual models and documentation, and reasoning support to help identify and validate casual scenarios. As such, using CAIRIS can to support your use of STPA could improve your efficiency as your analysis evolves.

One particular benefit of CAIRIS is its interoperability. For example, you may wish to rely on Excel to maintain a hazard list or other control structure data. Because Excel is machine readable by many scripting tools, it is comparatively easy to turn Excel spreadsheets into CAIRIS models, which can be incrementally imported into CAIRIS, or convert CAIRIS models back to Excel.

In the following sections, we describe how CAIRIS can help with the four steps of STPA. Like the rest of the CAIRIS documentation, please help us help you by [raising an issue](#) about anything unclear, inaccurate, or to raise request additional content you think might be useful.

25.2 Step 1: Define purpose of the analysis

25.2.1 Identifying losses

You can add losses by creating obstacles with a Loss category.

25.2.2 Identifying system-level hazards

You can add hazards by creating an obstacle with a Hazard category. To link hazards to losses, add a KAOS association where the head element is the loss obstacle, the tail element in the hazard obstacle, and an *And* association type to indicate that the refined hazard needs to be satisfied for the loss to be achieved.

25.2.3 Defining system-level constraints

System-level constraints can be modelled as goals in CAIRIS. To indicate that the constraint attends to a hazard, add a KAOS association where the head element is a hazard obstacle, the tail element is the goal representing the constraint, and the association type is *Resolve*, to indicate the constraint is necessary for preventing the hazard.

25.2.4 Refining the system-level hazards

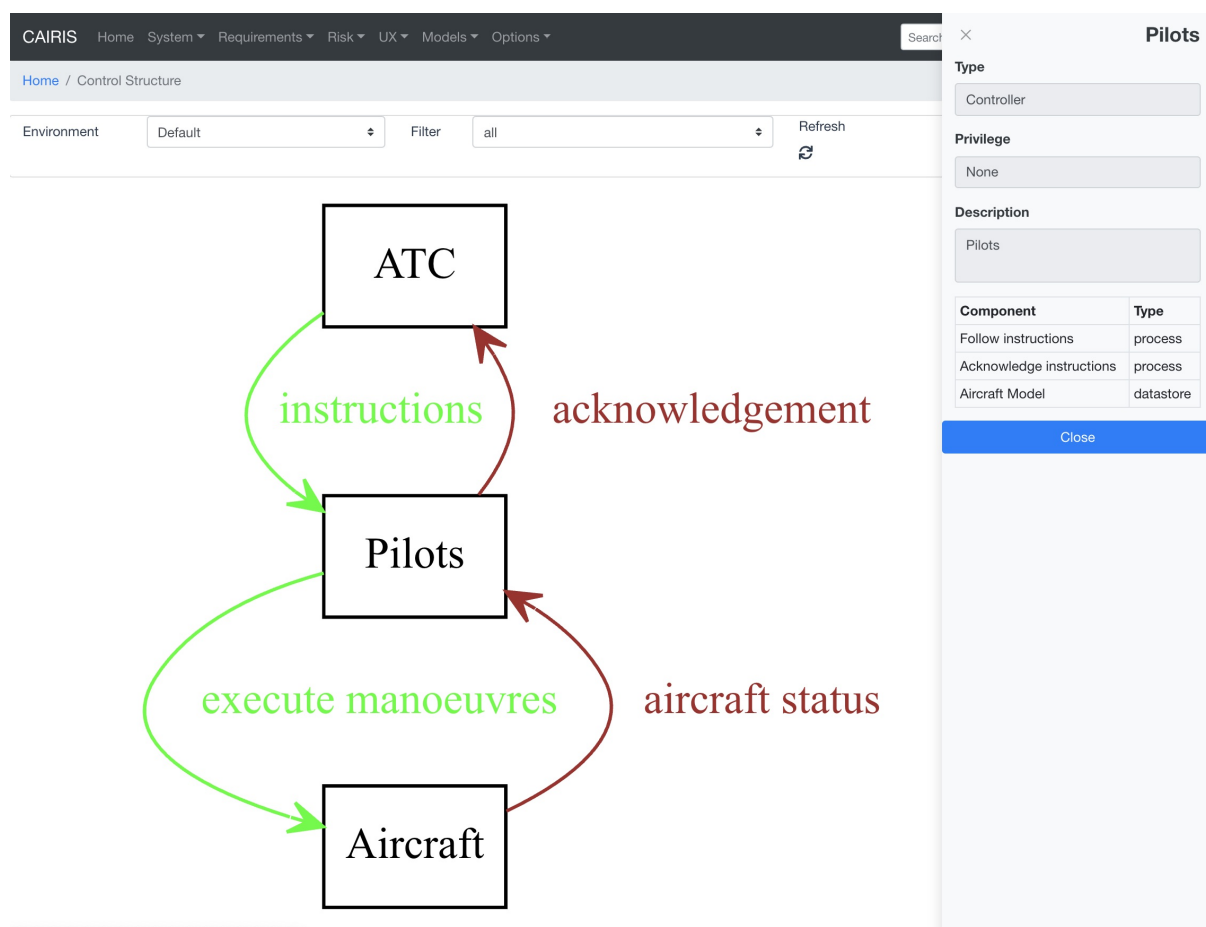
Additional obstacles and KAOS associations can model how hazards can be refined.

25.3 Step 2: Model the control structure

This entails creating a data flow diagram (DFD), where processes and data stores are analogous with STPA control algorithms and process models respectively. To create processes and data stores, you need to create use cases and information assets respectively. These DFD elements should be encompassed in a trust boundary; the type of trust boundary can be set to be a Controller, Controlled Process, Sensor or Actuator.

DFD elements interact with each other via data flows. When creating data flows, the type of these data flows can be set as Control, Feedback, or Information. You also need to specify information asset/s carried by each control, feedback, or information flow.

CAIRIS now supports a *control structure* model. As shown in the figure below, which is based on ATC example in [this STPA tutorial](#). CAIRIS automatically visualises the relationship between trust boundaries - in their role of controllers, controlled processes, sensor or actuators - and entities, which could represent external systems.



25.4 Step 3: Identify unsafe control actions

Unsafe control actions can be represented as obstacles. Once identified for a control flow, these can be associated with data flow. When associating the obstacle, you need to indicate the appropriate UCA keyword (does not provide, provides, provides too early, provides too late, provides out of order, stopped too soon, applied too long, not applicable) and provide some textual context for the unsafe action. In the future, we may add support for automatically generating these obstacles based on the data flow elements, keyword and context but, for now, this obstacle needs to be manually created.

Once the obstacle has been created, this can be linked with hazards using KAOS associations, where the head elements are hazard obstacle, and the tail elements are obstacle constituting the unsafe control actions.

25.5 Step 4: Identify loss scenarios

Tasks can be created and linked to hazards and system constraints using the KAOS associations, i.e. where the task is the tail element and obstacles and/or goals are head elements. Tasks might be used to illustrate why unsafe control actions occur, and why control actions could be improperly (or not) executed – possibly in the presence of safety constraints – leading to hazards.

CAIRIS model validation checks can highlight design-level issues that could lead to such scenarios. We are currently working on STPA specific validation checks, e.g. to identify control actions without feedback.

25.6 Supporting other STPA outputs

KAOS associations can be created to indicate system roles that are responsible for the satisfaction of goals (i.e. safety constraints).

CAIRIS can automatically generate requirement specifications from CAIRIS models. We are considering the idea of generating a more specific STPA specification document. We welcome [requests](#) for what its format should be.

Modelling access control needs and policies

26.1 Overview

Access control needs can be modelled in CAIRIS, together with access control policy statements. If both needs and policies are captured then model validation checks can identify potentially undesirable or insecure access possibilities.

26.2 Modelling access needs

Indicating the need *subjects* have for accessing *resources* is captured in asset associations, where subjects and resources are system, information, or people assets. The table below indicates permitted access needs for the different asset types. For example, a person might read some information, but not vice-versa. It may seem odd that information should be permitted to access resources but, during early stages of design, stakeholders might model some system that stores information as an information asset, or an information asset needs to access some resource because some out-of-scope system or person is handling that information, but capturing information-information access has some value.

Table 1: Permitted access needs between subjects (rows) and resources (columns)

	System	Information	People
System	Y	Y	N
Information	Y	Y	N
People	Y	Y	Y

Asset models visually model desired access between subjects and resources, and the ends of asset associations can capture access control information.

Possible adornments for access needs are one or more of read (r), write (w), and interact (x). The choice of *x* is inspired by the execute permission used in unix, but the use of the term *interact* allows for a wider range of interaction affordances than execution. Where r, w, or x adornment are not shown on the end of an association, it is assumed no access needs to take place.

For example, the figure above shows the relationship between a Works Diary and Diary Event. Both are information assets and the former contains one or more of the latter. The adornment on the tail end of the Works

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
test@test.com

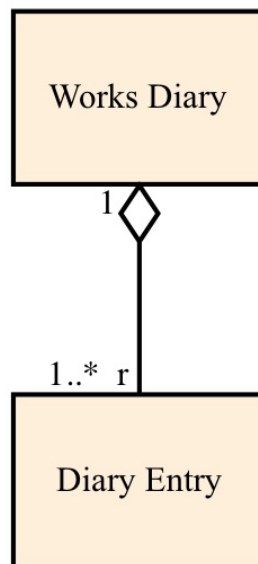
Home / Asset Model

Environment
Day

Asset
Works Diary

Hide Concerns
☒

Refresh



Diary-Diary Event association should be read as *a works diary needs to read a diary event*. The figure also indicates that diary events do not need to read, write, or interact with work diaries.

Multiple access needs can be captured in a comma-separated need list. For example, replacing *r* with *r,w* in the above example indicates that *a works diary needs to read and write a diary event*.

26.3 Modelling access control policies with policy statements

An access control policy captures the set of authorised and unauthorised interactions between assets. These interactions are captured using *policy statements*, where statement is defined as *Subject X Access X Resource X Permission*, where *Access* = {*read, write, interact*} and *Permission* = {*allow, deny*}.

A single KAOS goal is associated with a single policy statement. The goal needs to specify concern links to both the subject and resource assets. The goal should be precise enough to specify the conditions or capabilities the system needs to satisfy for the policy statement to hold. Where this is not possible, the goal needs to be refined. Consequently, a complete access control policy should correspond with a complete specification describing the intent the system needs to satisfy for compliance with the policy.

Access needs should correspond with goals and policy statements, but the absence of needs do not, i.e. adding deny policy statements. There are other means for capturing the rationale for non-inclusion, and goals may not be within the scope of analysis. This approach does not preclude the addition of goals and policy statements if they are.

If there is a goal and policy statement for interaction between a subject and resource, we would expect this to be refined to read and/or write interactions between the two.

- To add a policy statement, click on the Requirements / Policy Statement menu to open the Policy Statement table, and click on the Add button to open the Policy Statement form.
- Select the environment and goal with concern links to the subject and resource assets.

CAIRIS Home System Requirements Risk UX Models Options Search Search User

Home / Policy statements / Day / Diary entries / Works Diary / write / Diary Entry

Environment

Day

Goal

Diary entries

Subject Access Type Resource Permission

Works Diary write Diary Entry allow

Create Cancel

- Select the subject, access type, resource, and permission.
- Click on the Create button to add the new policy statement.
- Policy statements can be updated and deleted in a similar manner to other CAIRIS objects.

Note: At the time of writing, policy statements are not added to any generated documentation. When they are, this note will be removed from the documentation.

26.4 Access control model validation checks

A number of access control specific model validation checks are supported, e.g. No-read down violation checks for conflicting Integrity values. See [Access control checks](#) for a full list of supported checks.

CAIRIS Home System Requirements Risk UX Models Options Search Search test@test.com

Home / Validation Model

Environment

Day

Type	Description
Uncovered exception	Use case 'Modify Telemetry Software' is present in environment 'Day', but step 1 of the flow in this environment has an exception (No changes) with no related obstacle.
Implicit vulnerability	Information Security Manager depends on ICT Partner for goal Active Directory software deployment, but this goal is obstructed or denied.
No read-down violation	Works Diary needs read access to Diary Entry but reading down when the Integrity value of the subject is higher than the resource is undesirable

A risk is the detriment arising from an attacker launching an attack, in the form of a threat, exploiting a system weakness, in the form of a vulnerability. Associated with each risk is a misuse case. This describes how the attacker (or attackers) behind the risk's threat exploits the risk's vulnerability to realise the risk.

The current status of Risk Analysis can be quickly ascertained by viewing the Risk Analysis model. This displays the current risks, the artifacts contributing to the risk, and the artifacts which potentially mitigate it.

27.1 Adding, updating, and deleting a risk

- Select the Risk/Risks menu to open the Risks table, and click on the Add button to open the Risk dialog form.
- Enter a risk name and select a threat and vulnerability from the respective combo boxes. A risk is valid only if the threat and vulnerability exist within the same environment (or environments).
- Clicking on the environment name in the environment card populates the risk details card. The impact folder shows qualitative risk rating, and the mitigated and un-mitigated risk score associated with each risk response.
- Before a risk can be created, an associated Misuse Case needs to be defined. To do this, click on the Misuse Case folder.
- Most of the fields in the Misuse Case form have already been completed based on the risk analysis carried out up to this point. Enter a scenario which describes how the attacker realises the associated risk, i.e. carries out the threat by exploiting the vulnerability. The scenario written should be written in line with the attributes and values displayed.
- Click on the Create button to add the new risk.
- Existing risks can be modified by clicking on the risk in the Risks table, making the necessary changes, and clicking on the Update button.
- To delete a risk, click the Delete button next to the risk to be removed in the Risks table. If any artifacts are dependent on this risk then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the risk dependencies and the risk itself, or No to cancel the deletion.

CAIRIS
Home System Requirements Risk UX Models Options
 sfaily

Home / Risks / Unauthorised Certificate Access

Risk

Tags

Threat

Vulnerability

Environments

Psychosis

Stroke

Impact

Misuse Case

Rating

Name	Score (Pre Mitigation)	Score (Post Mitigation)	
Prevent Unauthorised Certificate Access	9	1	<input type="button" value="Show Details"/>

27.2 Risk Analysis model

Risk Analysis models show the contribution different design elements make to a risk, as shown below:

Risk Analysis models can be viewed by clicking on the Model/Risks Model menu, and selecting the environment to view the environment for.

By changing the environment name in the environment combo box, the risk analysis model for a different environment can be viewed.

The orientation of the model can be changed by layout radio button. By default, the orientation is set to Vertical but, if you are creating a kill chain that connects risk to threats or vulnerabilities, you might find a Horizontal layout helpful.

By clicking on a model element, information about that artifact can be viewed.

The risk analysis model can also be filtered by artifact type and artifact type. Filtering by type displays only the artifacts of the filtered type, and its directly associated assets. Filtering by artifact name displays only the filtered artifact, and its directly associated artifacts.

For details on how to print risk analysis models as SVG files, see [Generating Documentation](#).

CAIRIS
Home
System ▾
Requirements ▾
Risk ▾
UX ▾
Models ▾
Options ▾

sfaily ▾

Home / Risks / Unauthorised Certificate Access

Risk

Tags

Threat

Vulnerability

Environments

Psychosis

Stroke

Impact

Misuse Case

Attackers

Assets

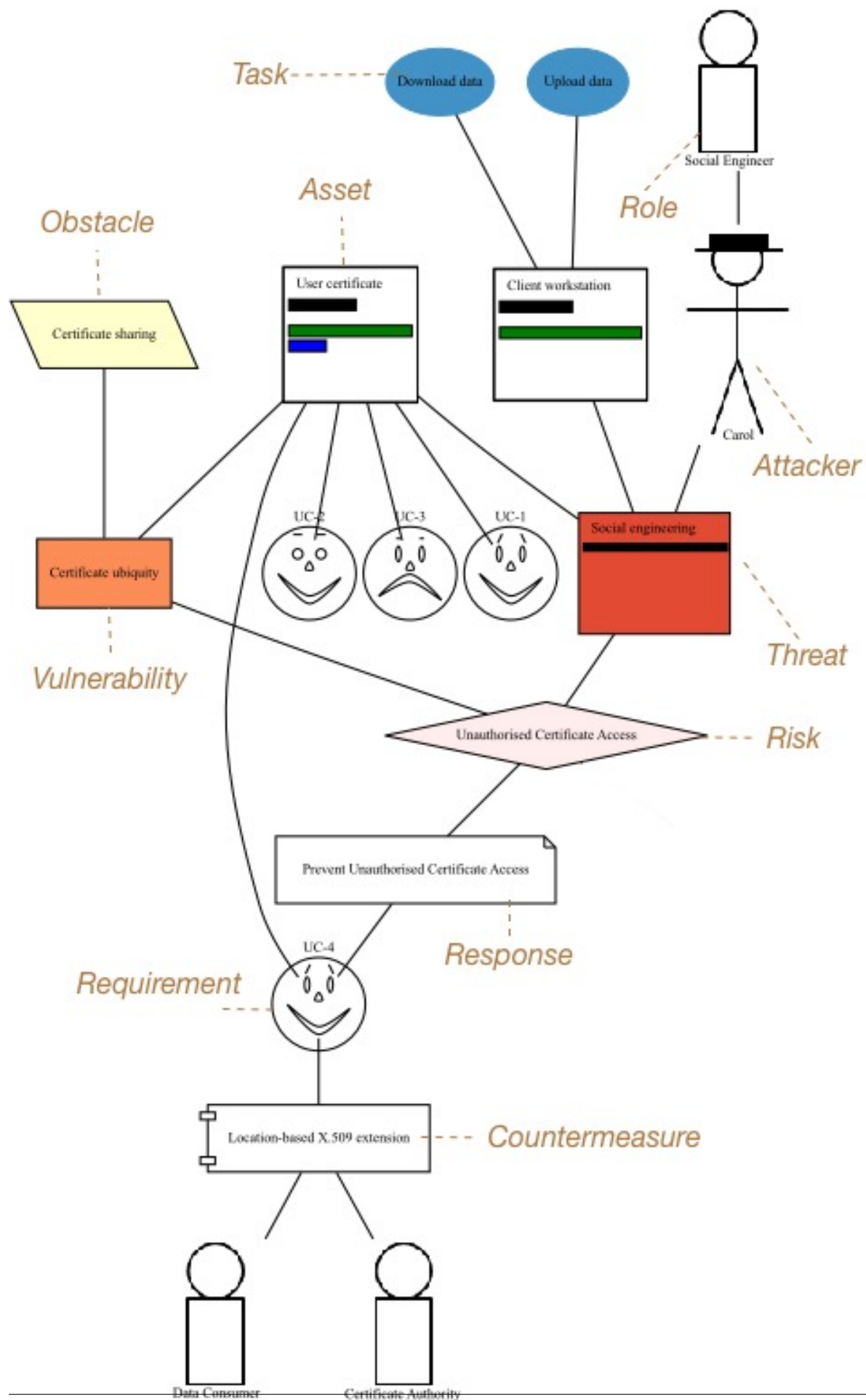
Objective

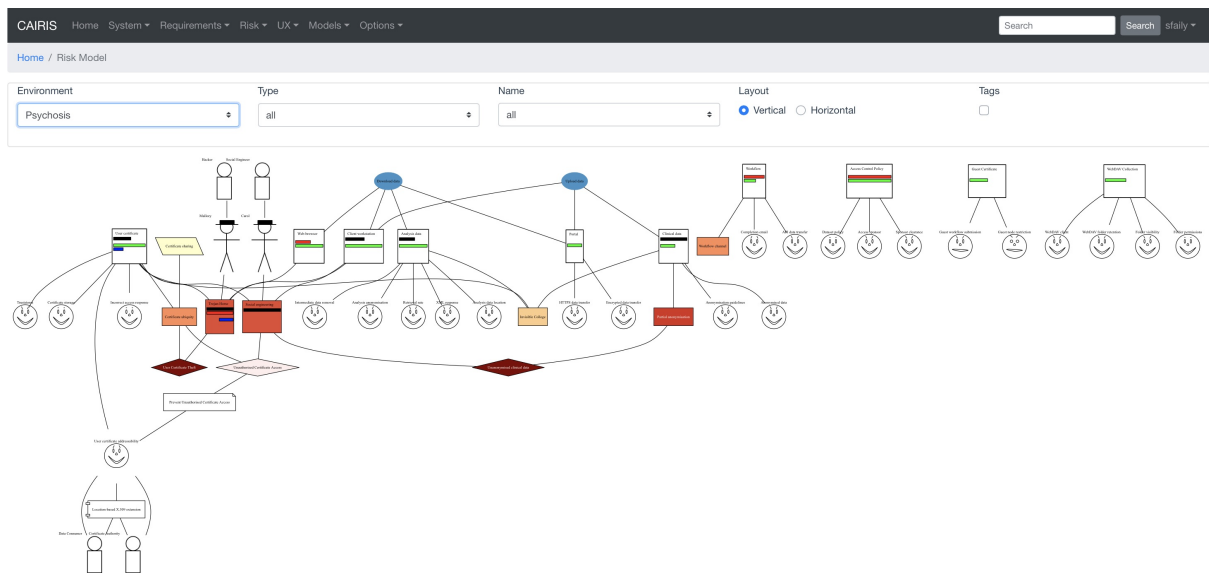
Likelihood

Severity

Narrative

Carol gained access to the Computing Laboratory by tailgating one of the many new DTC students arriving that day. When she reached the atrium she asked a nearby porter for directions to the Neurosciences areas. Carol followed the directions and eventually reached an open-plan area with a number of PCs. The computer room is a shared work area used by the many post-docs working in the department. Carol espied an unused PC, next to which was a man in his mid-20s who appeared to be deeply engrossed in his work. Carol approached the man and attempted to get his attention. "Hi there", Carol said, "I'm Carol, and I'm a new post-doc in the group". Dave looked up from his PC and smiled. "Hi", said the over-worked post-doc, "I'm Dave. Due mentioned something about a new post-doc. Well, I see you found your way here ok.". "Eventually", said Carol, "Oxford is a bit of a tardis isn't it. Is this desk free?". "Sure,



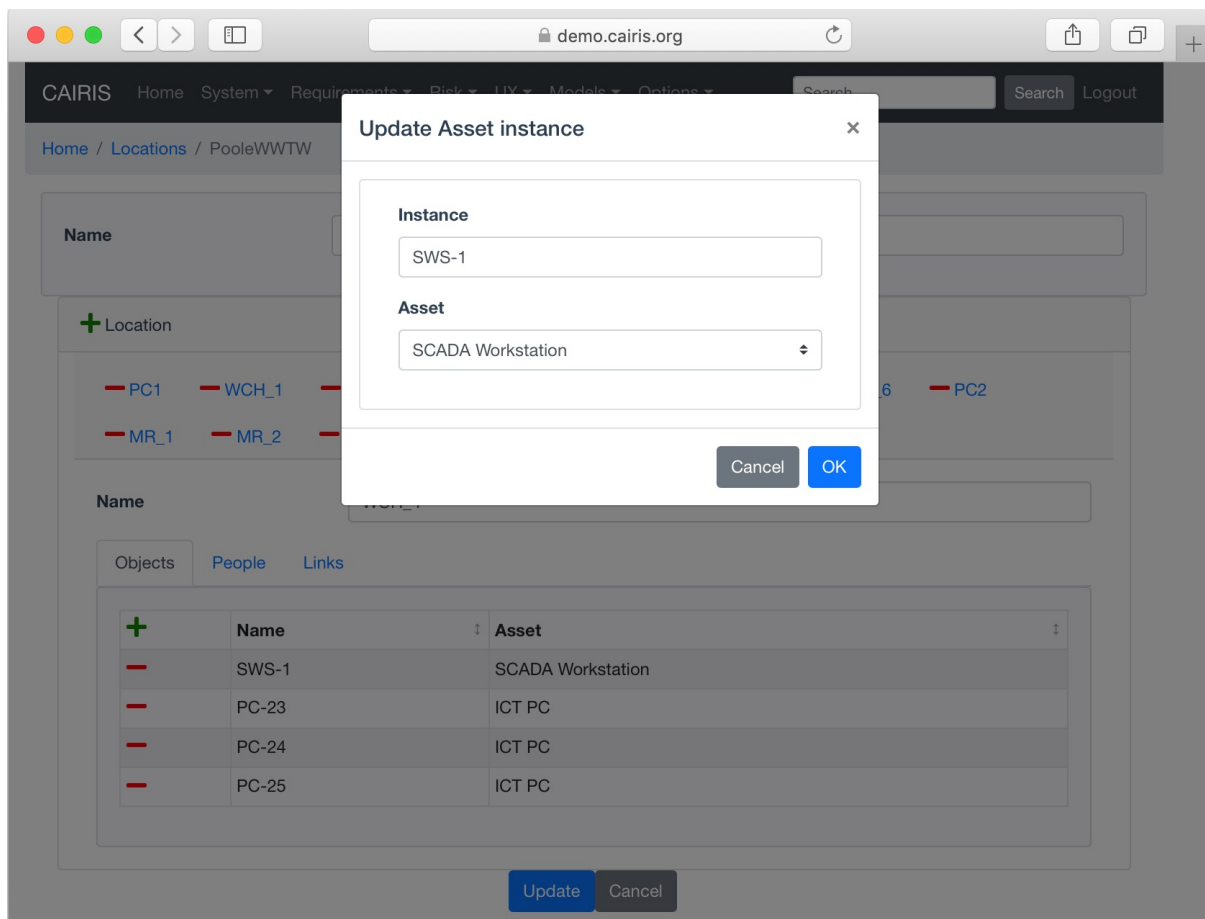


A Locations object is a collection of *location* objects connected by *links*. A location could be anything ranging from a room, corridor, or even a building. Within a *location* it is possible to specify instances of assets or personas. CAIRIS makes it possible to overlay risks onto location models, to explore the impact security might have on a physical location.

28.1 Adding, updating, and deleting a locations object

The screenshot shows the CAIRIS web application interface. The browser address bar displays 'demo.cairis.org'. The navigation menu includes 'CAIRIS', 'Home', 'System', 'Requirements', 'Risk', 'UX', 'Models', and 'Options'. A search bar and 'Logout' button are also present. The breadcrumb trail shows 'Home / Locations / PooleWWTW'. The main form is titled 'Name' and contains a text input field with 'PooleWWTW'. Below this is a '+ Location' button. A horizontal list of location instances is shown: PC1, WCH_1, WCH_2, WCH_3, WCH_4, WCH_5, WCH_6, PC2, MR_1, MR_2, MR_3, and MR_4. Below this list is another 'Name' input field with 'PC1'. There are three tabs: 'Objects', 'People', and 'Links'. The 'Objects' tab is active, showing a table with a '+ Add' button, a 'Name' column, and an 'Asset' column. At the bottom are 'Update' and 'Cancel' buttons.

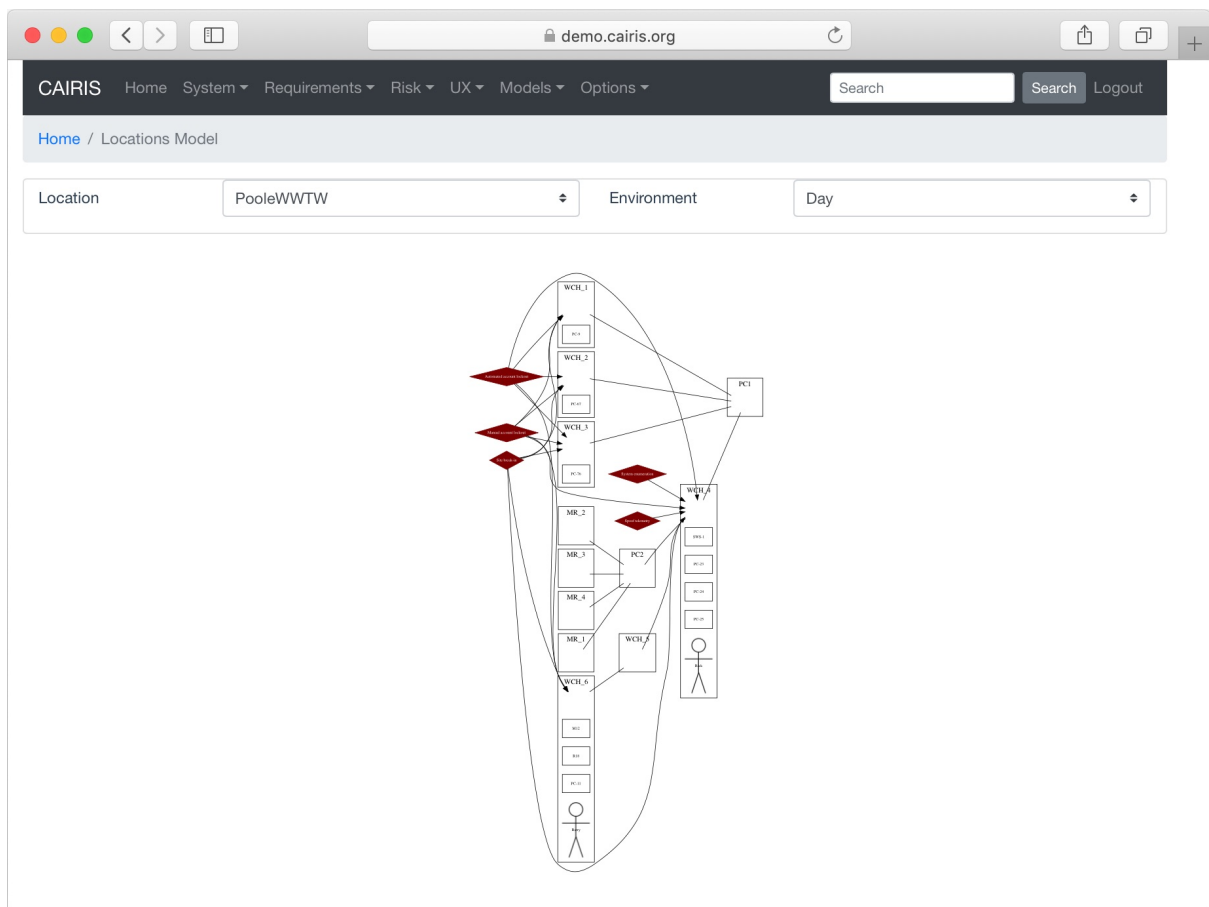
- Select the UX / Locations menu to view the list of Locations objects.
- Click on the Add button to open the Locations form, and enter the name of the Locations object.
- To add a location to this object, click on the Add button in the locations card.
- In the Objects folder, click on the Add button to open the Asset Instance dialog. Enter the name of the asset instance, select the asset name, and click on Add to add the instance object.
- In the Peopler folder, click on the Add button to open the Persona Instance dialog. Enter the name of the persona instance, select the persona name, and click on the Add button to add the instance object.



- In the Links folder, click on the Add button to open the Location Link dialog. Select the location you wish to link this location to, and click on the Add button to add the link between locations. When a link is added, a corresponding is added to the linked location.
- Click on the Update button to add this location to the Locations object.
- Existing Locations objects can be modified by clicking on the Locations name in the Locations table, clicking on individual location rows in the table of locations, and adding or deleting asset instances, persona instances, or links, before clicking on the Update button.

28.2 Viewing location models

Location models can be viewed by clicking on the Models/Locations menu option, selecting the Locations object and a specific environment.



Risk Responses

A risk can be treated in several ways.

By choosing to *Accept* a risk, we indicate we are prepared to accept the consequences of the risk being realised. Accepting the risk comes with a cost, and responsibility for accepting a risk must fall on one or more roles.

By choosing to *Transfer* a risk, we acknowledge that dealing with a risk is out of scope for this project. It may, however, have a cost associated with it and, by accepting the risk, the risk must become the responsibility of one or more roles.

By choosing to *Mitigate* a risk, we may either Prevent, Deter, Detect, or React to a risk. For detective responses, the response must detect the risk before, during, or after the risk's realisation. For reactive responses, the response must be associated with an countermeasure asset derived from a detective response.

29.1 Adding, updating, and deleting a response

- Select the Risk/Responses menu to open the Responses table, and click on the Add button. Select the response to take from the available options presented.
- Select the risk to associate this response with.
- Click on the Add button in the environment table, and select an environment to situate the threat in. This will add the new environment to the environment list.
- When the risk name and response type is selected, the response name is automatically generated.
- If an accept or transfer response was selected, a cost and rationale needs to be entered. For transfer responses, one or more roles also need to be associated with the response.
- If a Detect response is selected, select the Detection Point (Before, Medium, or After).
- If a React response is selected, Click on the Add button above the Detection Mechanism table, and select a detection mechanism asset.
- Click on the Create button to add the new response.
- Existing responses can be modified by clicking on the response in the Responses table, making the necessary changes, and clicking on the Update button.
- To delete a response, click the Delete button next to the response to be removed in the Responses table. If any artifacts are dependent on this response then a dialog box stating these dependencies are displayed. The

The screenshot shows a web browser window with the URL `demo.cairis.org`. The CAIRIS navigation bar includes links for Home, System, Requirements, Risk, UX, Models, and Options, along with a search bar and a Logout button. The breadcrumb trail indicates the current location: Home / Responses / Detect War-dialing attack.

The main configuration area for the 'Detect War-dialing attack' response includes the following fields:

- Type:** A radio button labeled 'Detect' is selected.
- Risk:** A dropdown menu showing 'War-dialing attack'.
- Response:** A button labeled 'Detect War-dialing attack'.
- Tags:** An empty text input field.

Below these fields is a section titled '+ Environment' which contains:

- A blue button labeled 'Night' with a red minus icon.
- Type:** Radio buttons for 'Prevent', 'Deter', 'Detect' (selected), and 'React'.
- Detection Point:** Radio buttons for 'Before', 'At', and 'After' (selected).

At the bottom of the configuration area are two buttons: 'Update' (blue) and 'Cancel' (grey).

user has the option of selecting Yes to remove the response dependencies and the response itself, or No to cancel the deletion.

29.2 Generating goals

A goal can be generated from a response by clicking on the Goal button in the responses table. This generates a goal in each of the environments the response is situated in. The goal name corresponds to the name of the response.

Countermeasures

After a response goal has been generated, goal modelling continues until one or more countermeasure requirements have been defined and associated with their parent goals. Following this, a countermeasure can be defined. Defining a countermeasure also has the effect of satisfying a response goal and resolving any obstacles associated with the underlying risk's threat or vulnerability.

Countermeasures target a risk's threat, vulnerability, or both. Countermeasures also have a level of effectiveness. This effectiveness level determines how much the countermeasure reduces the likelihood of the associated threat, or severity of the associated vulnerability.

Countermeasures are associated with roles, who may be responsible for developing, maintaining or using the countermeasure. Consequently, countermeasures are also associated with tasks and, when defining a countermeasure, it is also necessary to indicate how much the countermeasure helps or hinders the properties of associated tasks.

30.1 Adding, updating, and deleting a countermeasure

- Select the Risk/Countermeasures menu to open the Countermeasures form, and click on the Add button to open the Countermeasure form.
- Enter the countermeasure name and description, and select the countermeasure type. A countermeasure may be one of the following type: Information, Systems, Software, Hardware, or People.
- Click on the Add button in the environment card, and select an environment to situate the countermeasure in. This will add the new environment to the environment list.
- Select the countermeasure cost to indicate the general cost of implementing the countermeasure within the selected environment.
- Click on the Security tab to display the security page. Click on the Add button above the Requirements table, and select the requirement (or requirements) this countermeasure refines. Following this, click on the Add button above the Targets table to select the countermeasure's target/s, together with the countermeasure's effectiveness. Finally, add the security properties fostered by this countermeasure via the security properties box at the bottom of the page.
- Click on the Usability tab to display the usability page. Click on the Add button above the Roles table, and select the roles associated with this countermeasure. Any tasks associated with these roles are automatically populated in the Task box at the bottom of the page, together with the person/s carrying out the task. If the countermeasure helps or hinders a task, double click on the task and modify the task's attributes accordingly.

CAIRIS

HomeSystem▼Requirements▼Risk▼UX▼Models▼Options▼

Search

Search

Logout

Home / Countermeasures / Location-based X.509 extension

Countermeasure

Location-based X.509 extension

Type

Information

Description

X.509 certificates extended to tie client workstations so NeuroGrid tasks can only be carried out on these.

Tags

+ Environment

Psychosis

Stroke

Cost

Low

Medium

High

Security

Usability

+ Requirement

User certificate

+ Name

Certificate ubiquity

Effectiveness

High

Rationale

Discourages certificate sharing

+ Property

Value

Rationale

Update

Cancel

128

Chapter 30. Countermeasures

The screenshot shows a web browser window with the URL `demo.cairis.org`. The application header includes the CAIRIS logo and navigation links: Home, System, Requirements, Risk, UX, Models, and Options. A search bar and a Logout button are also present. The breadcrumb trail indicates the current location: Home / Countermeasures / Location-based X.509 extension.

The main form for the 'Location-based X.509 extension' countermeasure is displayed. It includes a 'Countermeasure' field with the value 'Location-based X.509 extension' and a 'Type' dropdown menu set to 'Information'. The 'Description' field contains the text: 'X.509 certificates extended to tie client workstations so NeuroGrid tasks can only be carried out on these.' There is also a 'Tags' field.

Below the form, the 'Environment' section is shown. It includes a '+ Environment' button, a 'Psychosis' button (highlighted in blue), and a 'Stroke' button. The 'Cost' section has three radio buttons: 'Low', 'Medium' (selected), and 'High'. The 'Security' and 'Usability' tabs are visible, with 'Usability' currently active.

The 'Usability' tab displays a table with the following data:

Task	Persona	Duration	Frequency	Demands	Goal Conflict
Upload data	Claire	None	None	None	Low Hindrance
Download data	Claire	None	None	None	Low Hindrance

At the bottom of the form, there are 'Update' and 'Cancel' buttons.

- Click on the Create button to add the new countermeasure.
- Existing countermeasures can be modified by clicking on the countermeasure in the Countermeasures table, making the necessary changes, and clicking on the Update button.
- To delete a countermeasure, click the Delete button next to the countermeasure to be removed in the Countermeasures table. If any artifacts are dependent on this countermeasure then a dialog box stating these dependencies are displayed. The user has the option of selecting Yes to remove the countermeasure dependencies and the countermeasure itself, or No to cancel the deletion.

30.2 Generating countermeasure assets and security patterns

By clicking on the situate button in the countermeasures list, an associated asset can be generated. If defined, this will retain the same security properties associated with the countermeasure. The asset will be situated in whatever environments the countermeasure was situated in. In the asset model, a << safeguard >> association is added between the countermeasure asset and any assets threatened or exposed by the risk the countermeasure helps mitigate.

Assets can be generated directly based on the countermeasure properties, or on the basis of a pre-existing template asset. It is also possible to situate security patterns based on a countermeasure, rather than an asset.

Security Patterns can be imported into the tool by selecting the System/Import Model menu option and, when selecting the XML model file to be imported, selecting Security Pattern option. An example catalogue file, `schumacher_patterns.xml`, which incorporates a number of patterns from the Security Patterns text book by Schumacher et al is included in the `cairis/examples/architecture` directory.

30.3 Associating countermeasures with pre-existing patterns

In the situate form, you can also associate a countermeasure with a pre-existing security pattern. However, a list of possible security patterns to choose from will only be displayed if the components of the security pattern are present in ALL of the environments the countermeasure is situated for.

30.4 Weakening the effectiveness of countermeasures

Countermeasures mitigate risks by targeting its risk elements, i.e. its threats or vulnerabilities. However, when one or more assets are generated from these countermeasures, several factors may weaken the effect of the countermeasure.

First, situating assets may cause you to look at the environments where the assets are situated in a different light. Changing properties of assets, or existing threats or vulnerabilities could increase the potency of the risk, thereby weakening the effect of the countermeasure.

Existing threats or vulnerabilities can also explicitly weaken countermeasures. If a countermeasure asset is associated with a threat or vulnerability then, when either artifact is created or modified, CAIRIS allows users to override the effectiveness of the related countermeasure. The detail associated with the risk scores in the Risk Dialog box will indicate cases where countermeasures have been weakened by threats and/or vulnerabilities.

30.5 Mitigating weakening effects

If a countermeasure is weakened, the weakness is removed by generating a new countermeasure which targets the weakening threat or vulnerability. If this is carried out, the detail associated with the risk score in the Risk Dialog box will indicate cases where, although the effectiveness score for the countermeasure holds, this is by virtue of a countermeasure targeting the weakening threat or vulnerability.

Countermeasures cannot, however, be simply defined on the fly. They arise as the result of rational risk analysis, so risks need to be defined based on the weakening threats or vulnerabilities.

31.1 Allowable manual traceability links

CAIRIS is based on the IRIS meta-model. In most cases, traceability between model elements is automatic because the CAIRIS database knows how model elements are connected based on this meta-model. In some cases, however, it is necessary to add manual traceability relationships, e.g. between one requirement and other.

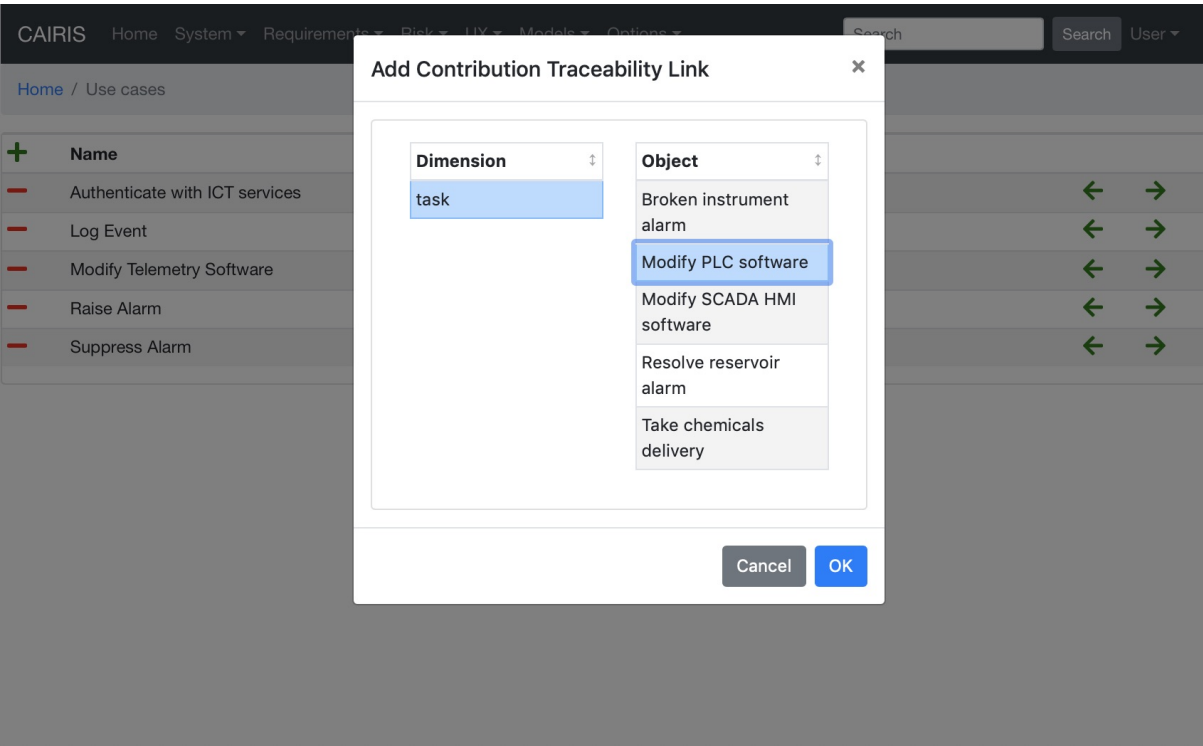
The table below indicates what manual links are allowed to be set between elements.

From	To
Requirement	Task
Task	Vulnerability
Requirement	Vulnerability
Asset	Requirement
Requirement	Role
Requirement	Use Case
Use Case	Task
Requirement	Requirement
Requirement	Document Reference
Risk	Threat
Risk	Vulnerability
Component	Use Case
Document Reference	Vulnerability
Document Reference	Obstacle

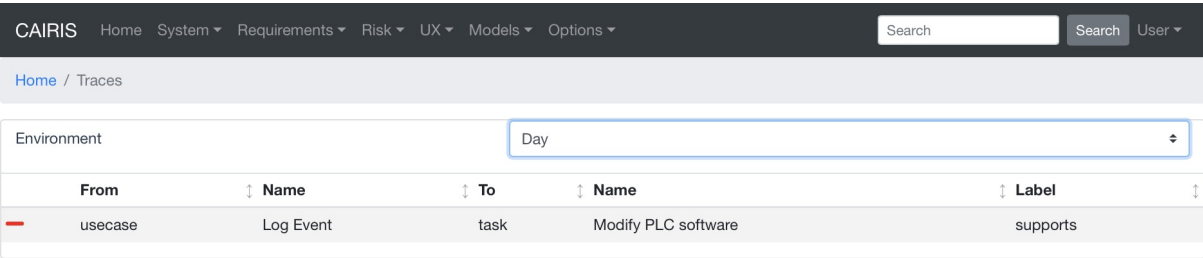
31.2 Editing manual traceability links

To add manual traceability links, right click on the left arrow for a *Supported by* (pre-traceability) link, or the right arrow for a *Contributes to* (post-traceability link). This will open a modal box for adding the forward or backwards traceability link.

If the traceability link is between two requirements, you will also be able to add a label describing the nature of the traceability.

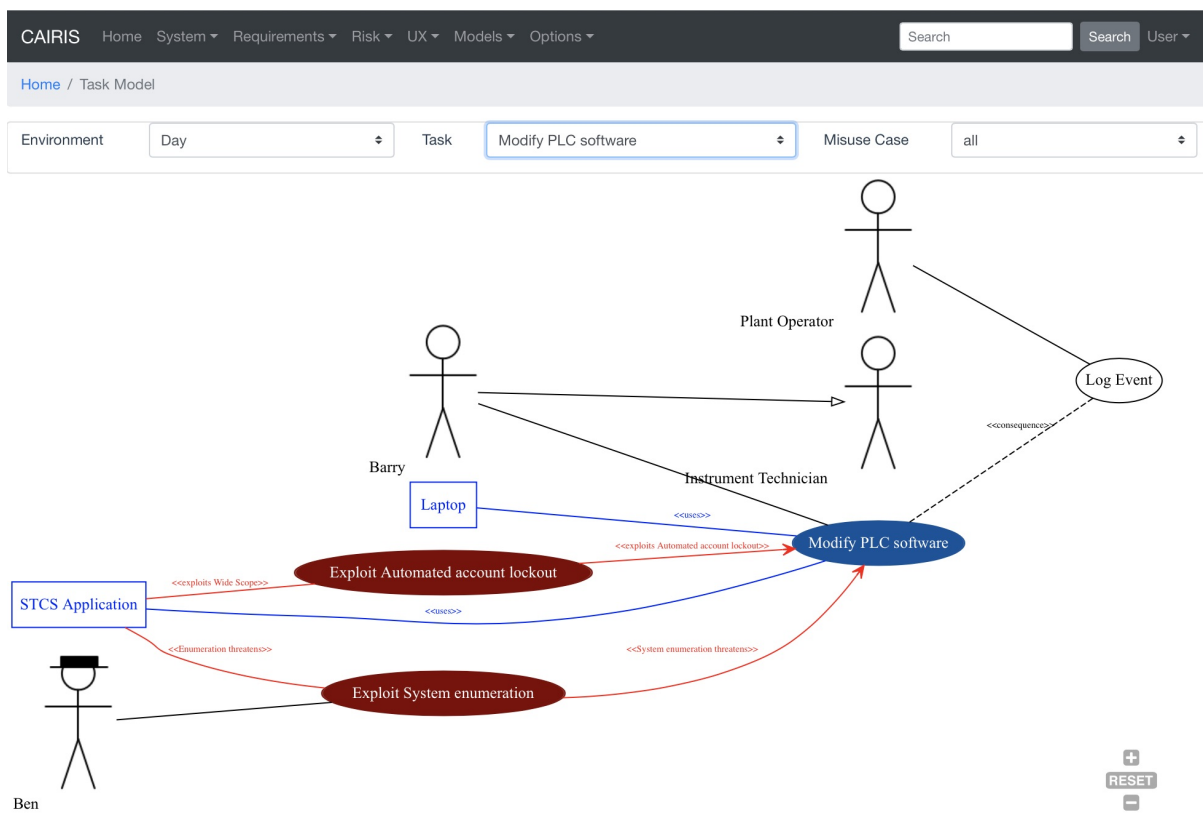


To delete a manual traceability link, select the Options / Traceability menu, select the environment that the traceability link is specific to if appropriate, and select the delete icon.



31.3 Visualising manual traceability links

Manual traceability links might be visualised in different ways. For example, in this example, a consequences link is added between the Log Event use cases and Modify PLC software task, which is visible on task models.

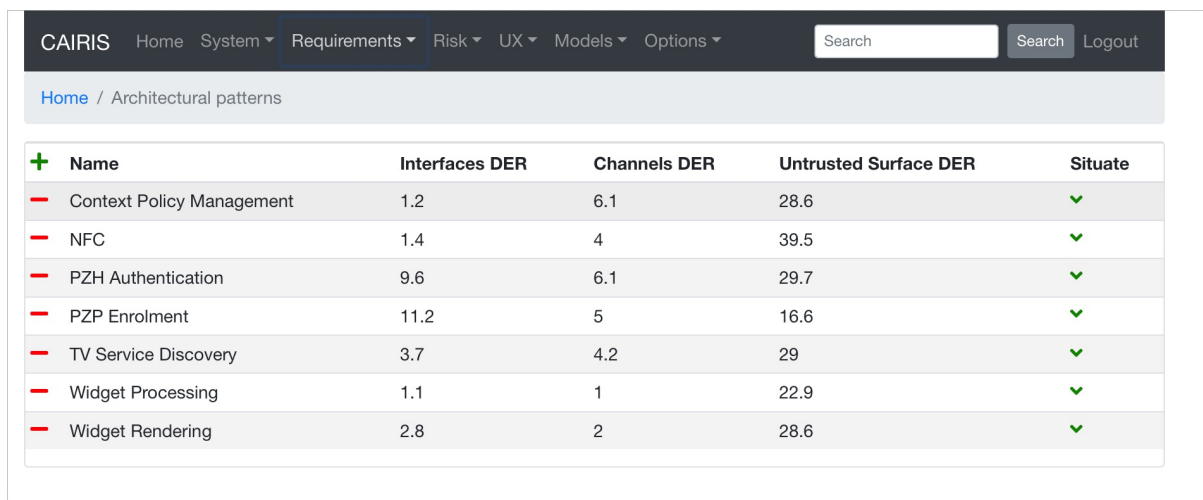


Architectural Patterns
















Designing software doesn't start with a blank page, but with a bricolage of different model elements.

Architectural patterns model pre-defined components, connectors, responsibilities, and requirements or goals that the model elements satisfy.

32.1 Editing Architectural Patterns



The screenshot shows the CAIRIS web application interface. At the top is a navigation bar with links: CAIRIS, Home, System, Requirements, Risk, UX, Models, and Options. There is also a search bar and a Logout button. Below the navigation bar is a breadcrumb trail: Home / Architectural patterns. The main content area displays a table of Architectural Patterns. The table has five columns: Name, Interfaces DER, Channels DER, Untrusted Surface DER, and Situate. There are eight rows of data, each with a red minus icon in the first column and a green checkmark in the last column.

 Name	Interfaces DER	Channels DER	Untrusted Surface DER	Situate
 Context Policy Management	1.2	6.1	28.6	
 NFC	1.4	4	39.5	
 PZH Authentication	9.6	6.1	29.7	
 PZP Enrolment	11.2	5	16.6	
 TV Service Discovery	3.7	4.2	29	
 Widget Processing	1.1	1	22.9	
 Widget Rendering	2.8	2	28.6	

- Select the Requirements / Architectural Patterns table to view the table of Architectural Patterns. The table provides a summary of the Damage-Effort Ratios for Interfaces, Channels, and Untrusted Surfaces. These metrics are explained in more detail [here](#).
- To create a new Architectural Pattern in CAIRIS, click on the Add button to open the Architectural Pattern form.
- Enter the name for the architectural pattern and provide a synopsis for the pattern.
- Select the Component tab, click on the Add button to add a new Component folder for the component being created.
- Enter a name for component.

CAIRIS
Home
System
Requirements
Risk
UX
Models
Options
Search
Search
Logout

Home / Architectural patterns / Context Policy Management

Architectural Pattern
Context Policy Management

Synopsis
Model illustrating how policy management mediates the

+ Component
Connectors

Policy Manager
Context Manager
webinos API
Context API
Context Database

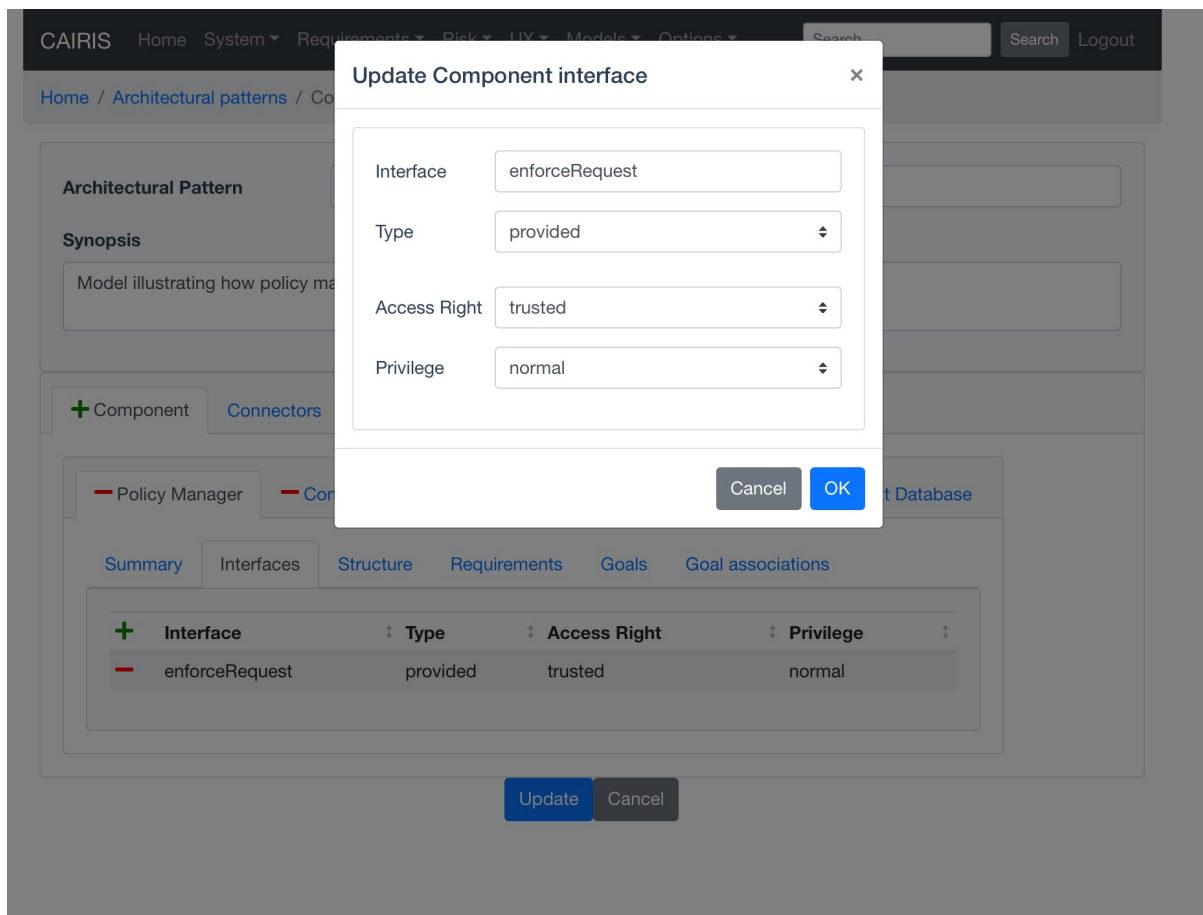
Summary
Interfaces
Structure
Requirements
Goals
Goal associations

Name
Policy Manager

Description
XACML based policy manager component.

Update
Cancel

- Select the Interfaces folder, and click on the Add button to add a new interface.
- Enter the name of the interface, and select whether the interface is provided or required, the interface access right, and privilege. Click on Add to add the interface to the component.
- Select the Structure folder, and click on the Add button to add a new template asset association to the component.
- For both the head and tail end of the template asset association, select the template asset, navigability indicator (0 or 1), adornment (inheritance, association, aggregation, composition, or dependency), cardinality *nry* (1, , or *I..*), and an optional role. Click on Add to add the association to the component template asset structure.
- If there are template requirements associated with the component, click on the Requirements folder and click on the Add button to select a requirement, and click on the Add button to add the template requirement to the component.
- If there are template goals associated with the component, click on the Goals folder and click on the Add button to select a goal, and click on the Add button to add the template goal to the component.
- To add relationships between template goals associated with the component, click on the Goal Associations folder and click on the Add button to add a new Goal Association.
- Select the template goal and sub-goals, the form of refinement (and / or), and enter some rationale for this relationship. Click on the Add button to add this goal association to the component.
- Click on the Update button to add the component to the architectural pattern.
- Select the Connectors folder, and click on the Add button to open the Connector form.
- Enter a name for the connector.



- Select the From folder to enter details of the *from* end of the connector. Enter the role name, and select the component and component interface.
- Select the To folder to enter details of the *to* end of the connector. Enter the role name, and select the component and component interface.
- Select the Details folder to enter information about the connector itself. Select the asset being carried by the connector, the connector protocol, and the access right needed to interact with the connector. Available assets are drawn from the asset structure of both components.
- Click on the Update button to add the connector to the architectural pattern.
- Click on the Create button to add the architectural pattern.

32.2 Viewing Architectural Patterns

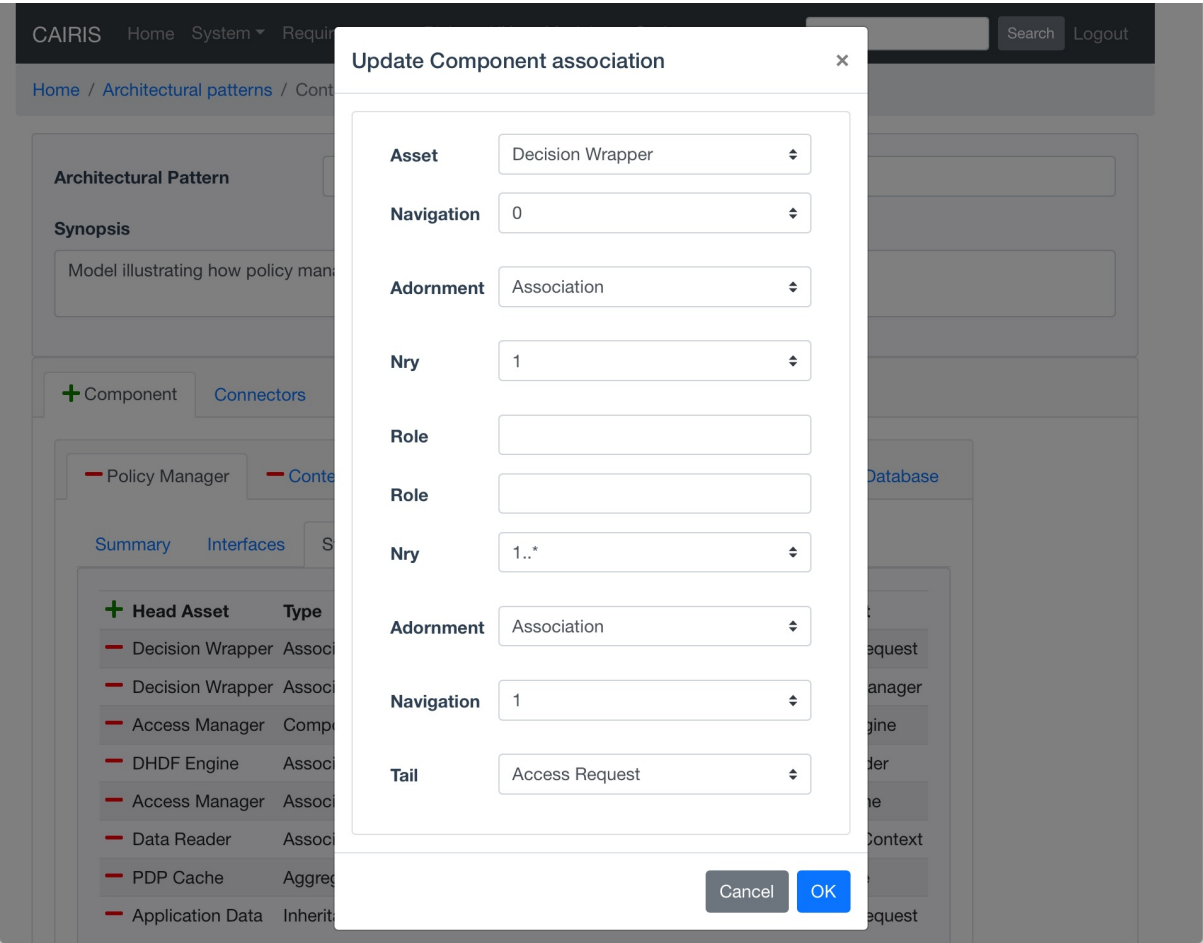
Architectural patterns can be viewed by clicking on the Models/Architectural Patterns menu option, and selecting the architectural pattern to display

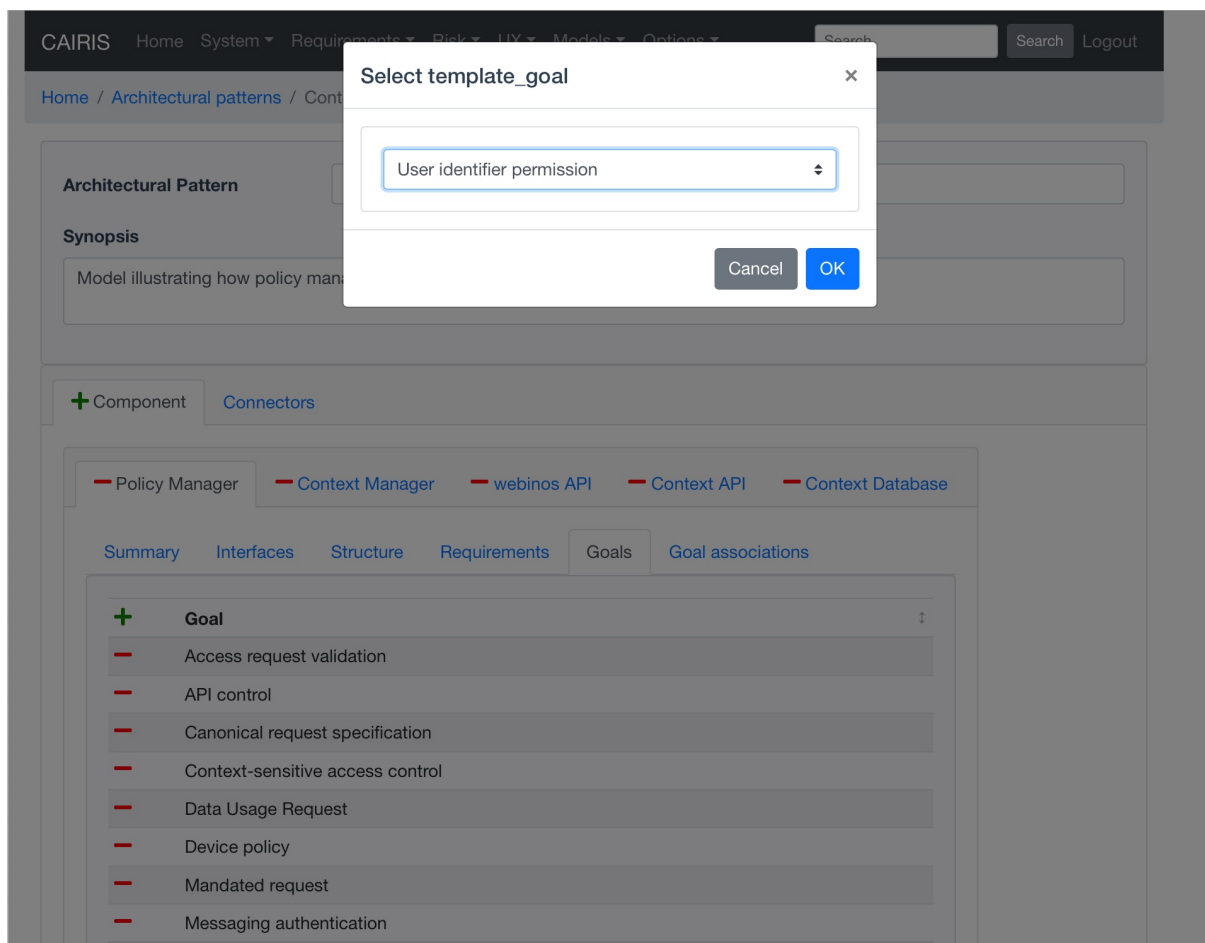
Components in this model are shaded red based on the component attack surface, This value is based on the DER_i metric.

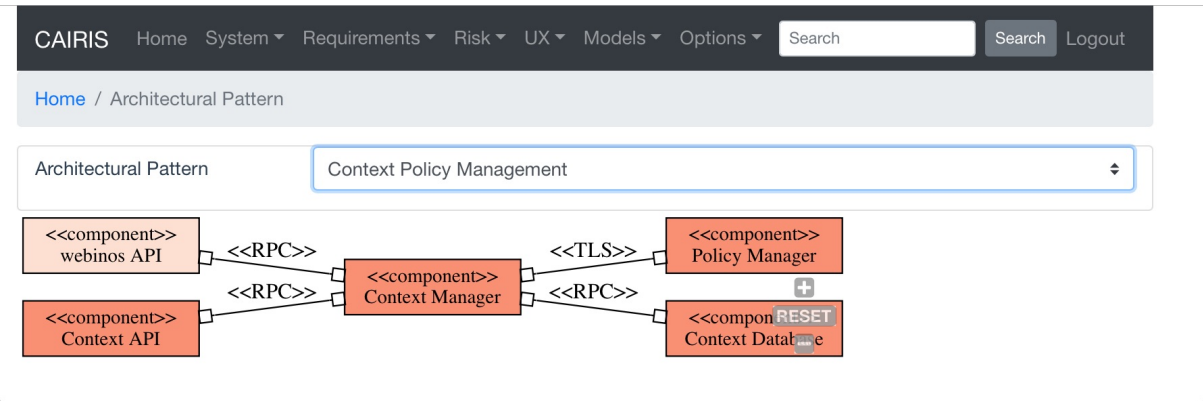
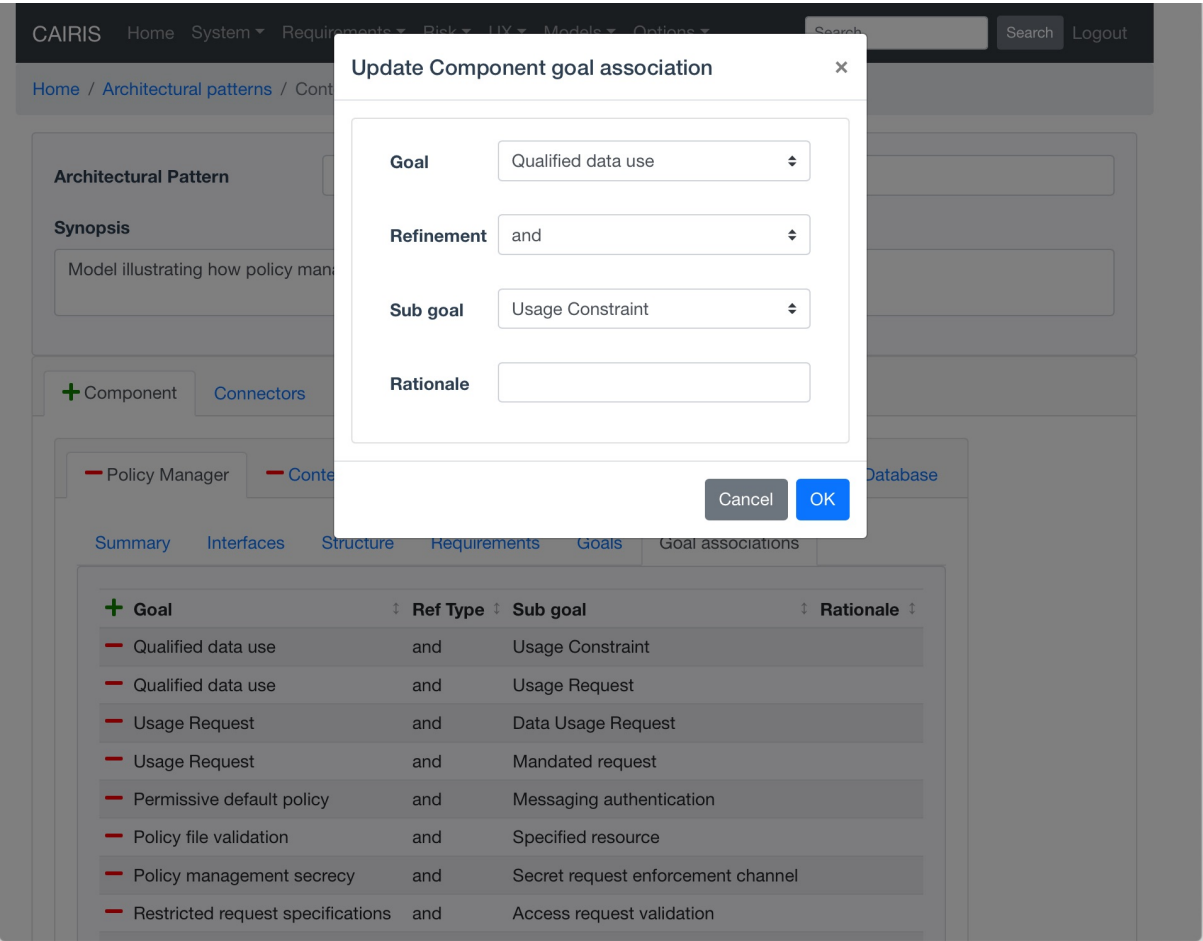
It is also possible to view asset and goal models associated with a component in an architectural pattern by clicking on the Models/Component Asset and Models/Component Goal respectively.

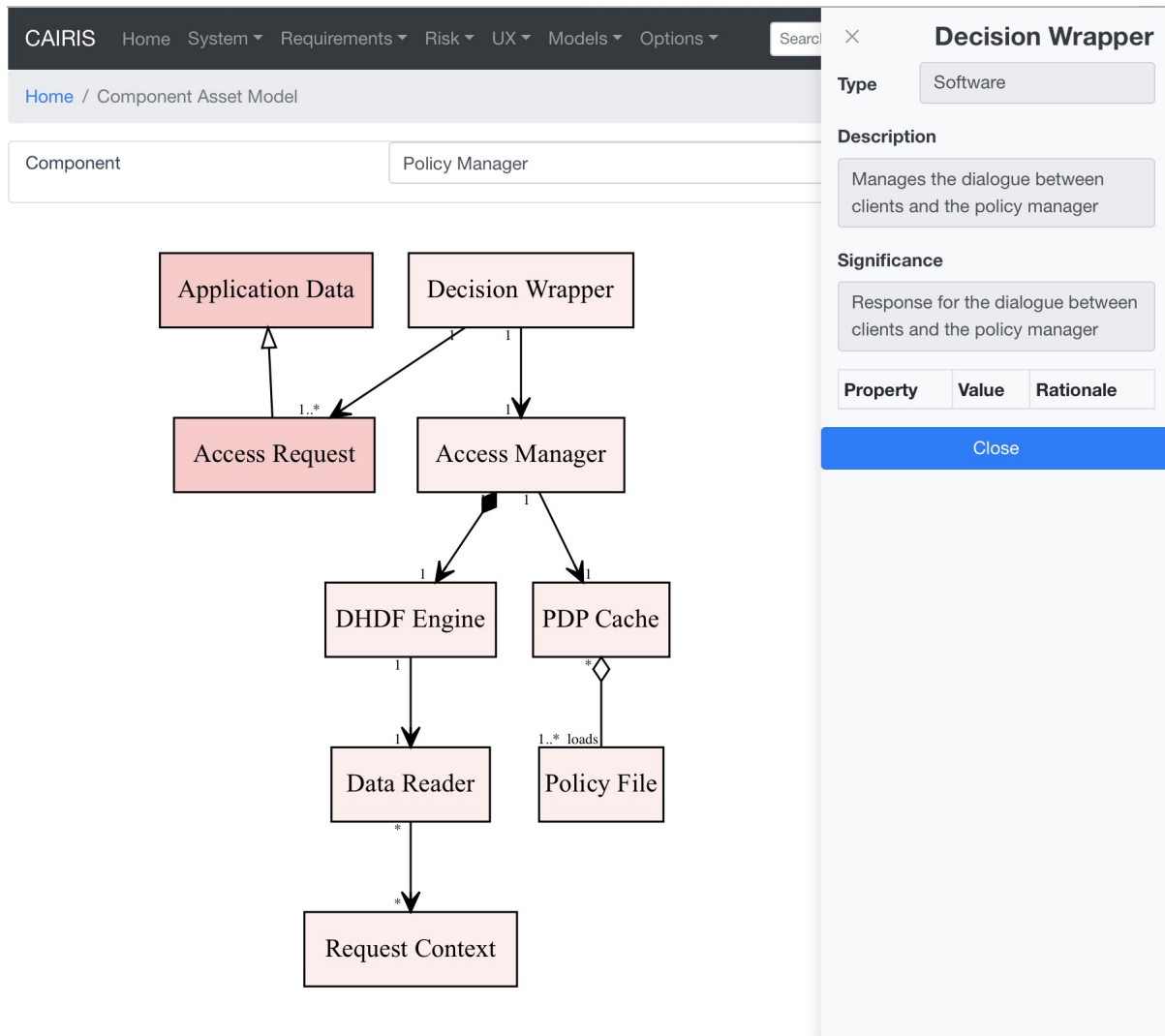
32.3 Situating a pattern

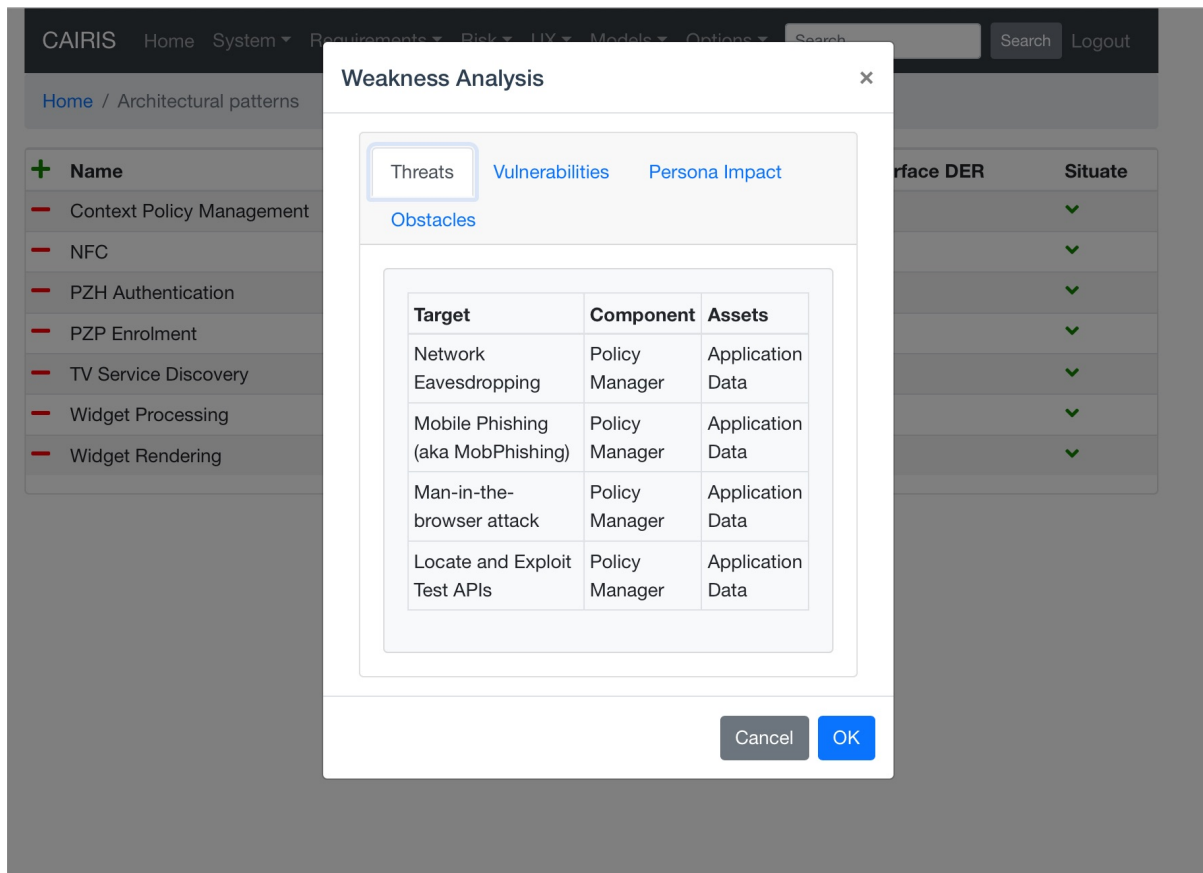
- To introduce an architectural pattern into the working project.











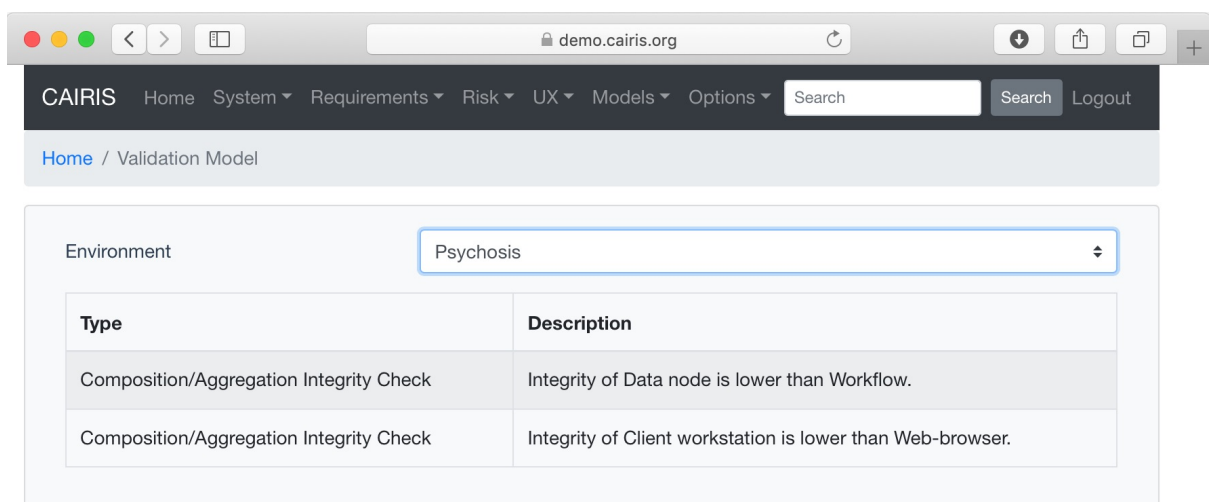
- Click on the situate button next to the desired architectural pattern in the architectural patterns table, followed by the environment to situate the pattern in. A weakness analysis dialog is displayed, which summarises the threats and vulnerabilities that will become associated with the pattern, the impact to persona task scores, and obstacles mitigated by goals introduced by the pattern.
- Click on the Ok button introduces the patterns template goals, requirements, and assets as standard goals, requirements, and assets respectively.

CHAPTER 33

Model Validation

CAIRIS can validate models for a given environment based on potential security and privacy design problems.

To validate a current CAIRIS model, click on the Models/Validate menu and select the environment to check the CAIRIS model for.



33.1 General validity checks

The general validity checks currently supported are as follows:

Check	Description
Reserved characters	Check object names for the presence of reserved characters.

33.2 Security design checks

The security design checks currently supported are as follows:

Check	Description
Composition/Aggregation Integrity	For Hardware/Software/Information assets, checks asset integrity for the head asset isn't lower than the tail asset.
Implicit asset inclusion	Asset implicitly included in an environment because of an asset association but no security or privacy properties have been set.
Implicit vulnerability	Checks whether a goal dependency is obstructed or a related user goal is denied, thereby introducing a vulnerability due to goal non-fulfilment.
Inherited asset inconsistency	Checks an asset inheriting from another asset doesn't have weaker security or privacy properties.
Inherited asset type	Checks an asset inheriting from another asset have the same asset type.
New risk contexts	Risks present in environments that haven't been accounted for.
Obstructed tasks	Where goals operationalise tasks, check root goals can be satisfied if any of its refined goals are obstructed.
STPA: potential control action conflict	Checks if multiple control flows feed into controlled processes.
Uncovered exception	Exception present in use case without a related obstacle.
Vulnerable non-valued asset	Asset is vulnerable but no security or privacy properties have been set for it.

33.3 Privacy design checks

If personal data has been introduced then the CAIRIS model is checked to ensure it doesn't violate any General Data Protection Regulation (GDPR) principles. The checks carried out are described below:

GDPR Principle	Check	Description
Lawfulness, Fairness, and Transparency	Fair data processing	Data with privacy properties is processed only if it's recognised as personal data.
Lawfulness, Fairness, and Transparency	Lawful data handling	A persona working with a task or use case involving personal data is a Data Processor, Data Controller, or Data Subject.
Lawfulness, Fairness, and Transparency	Necessary processing	Use cases involving personal data are associated with a necessary goal or requirement.
Purpose Limitation	Data purpose	Use cases involving personal data are associated with a necessary goal concerned with that personal data.
Data Minimisation	Private data processing	Data with privacy properties are accounted for in processes.
Accuracy	Personal data integrity	Personal data has an Integrity security property.
Storage Limitation	Unprocessed personal data	Personal data in data stores is processed.
Integrity & Confidentiality	Unmitigated privacy risks	Personal information has confidentiality, integrity, and privacy properties that threats target are not exposed to unmitigated risks.

33.4 Access control checks

These checks occur only if access needs and policy statements have been defined.

Check	Description
Unauthorised access	Subject needs access to a resource, but this access is denied in a policy statement.
Absent policy statement	Subject needs access to a resource, but no policy statement specifies this access.
Ambiguous policy statement	Subject needs access to a resource, but multiple policy statements specify this access.
No read-up violation	Subject needs access to a resource, but reading up when the Confidentiality value of the resource is higher than the subject is undesirable.
No write-down violation	Subject needs access to a resource, but writing down when the Confidentiality value of the subject is higher than the resource is undesirable.
No read-down violation	Subject needs access to a resource, but reading down when the Integrity value of the subject is higher than the resource is undesirable.
No write-up violation	Subject needs access to a resource, but writing up when the Integrity value of the resource is higher than the subject is undesirable.
No interaction up violation	Subject needs access to a resource, but interacting up when the Integrity value of the subject is lower than the resource is undesirable.

Configurable Types and Values

34.1 Asset Values

You can assign descriptions for None, Low, Medium, and High asset values by selecting the Options / Asset Value value menu, and clicking on the Value to be updated.

34.2 Asset Types

By default, CAIRIS databases are pre-defined with Information, Systems, Software, Hardware, and People asset types.

You can add a new asset type by selecting the Options / Asset Types menu and clicking on the Add button. You should then enter an asset type name and description, before clicking the Create button.

Existing asset type name and descriptions can be modified by clicking on an asset type to be updated, making the required changes, and clicking on Update.

A selected asset type can be deleted by clicking on its delete button. Although possible, we do not recommend deleting the standard asset types that come with CAIRIS.

If you are modelling a System of Systems, you may also benefit from using the asset type of *Systems - General*, used to represent organisations, groups, or social systems, and asset type *System of Systems* representing the combination of Independent Systems (collaborating to achieve a new combined purpose and goal).

34.3 Vulnerability and Threat Types

By default, CAIRIS models are pre-configured with vulnerability and threat types in the [ICS Protection Profile](#). These can be updated or overwritten, e.g. using one of the other threat and vulnerability type XML models.

34.4 Other Types

You can add a selection of types associated with risk and architectural models from the appropriate Options sub-menu by clicking on the Add button. You should then enter a name, score, description, and – where appropriate – score and rationale before clicking on the Create button.

Existing values can be modified by clicking on the to be updated, making the required changes, and clicking the Update button.

A selected type can be deleted by clicking on its delete button.

Searching model objects

It is possible to search for a requirement or any other model object with a particular text string from the Search box in the menu bar. Entering text will search the CAIRIS database and return a table of model elements where the text is present. Clicking on the selected row in this table will open the associated model object.

CAIRIS				Home	System ▾	Requirements ▾	Risk ▾	UX ▾	Models ▾	Options ▾	SCADA	Search	test ▾
Home / Find Model													
Environment	Dimension	Object											
	Project Settings	Project Goals											
	Project Settings	Project Scope											
	Role	Vendor											
	Role	Instrument Technician											
	Persona Characteristic	React to alarms raised by SCADA											
	Persona Characteristic	Readings periodically taken from SCADA											
	Persona Characteristic	SCADA isolation makes hacking unlikely											
	Persona Characteristic	Concern about EnterpriseSCADA uncertainty											
	Document Reference	SCADA PCs are site-specific industrial PCs.											
	Document Reference	SCADA authentication via non ACME credentials											
	Document Reference	Readings are taken from SCADA screens.											
	Document Reference	PC and SCADA security											
	Document Reference	Stand-alone SCADA											
	Document Reference	Uncertainty about EnterpriseSCADA ability to support general operations.											
	Document Reference	Concern about lack of training on EnterpriseSCADA.											
	Document Reference	Little knowledge about long-term EnterpriseSCADA impact											
	Document Reference	No SCADA logouts											
	Document Reference	Fixes problems identified by SCADA											
	Document Reference	Few SCADA to EnterpriseSCADA process changes											
	Document Reference	Hacking indifference											
	Document Reference	Migration pending resources											
	Document Reference	Incompatible networks											

Tags

Most objects in CAIRIS can be assigned one or more *tags* to categorise them. Tags can aid searching, i.e. by searching for objects where the tag has been set.

These tags can be entered in the Tags field in the appropriate form. Multiple tags can be added by separate individual tags with commas.

Tags can also be used to group objects in risk models. The example below shows a risk model with a *kill chain* categorised by [ATT&CK](#) tactic.

Some assets might represent entities in DFDs and, when it does and where appropriate, we can use asset tags to indicate roles that are synonymous with assets. For example, if the people asset of *Technician* is synonymous with the role *Instrument Technician*, you can indicate this by setting the asset tag as *role=Instrument Technician*.

If, in the asset model, the persona fulfils a role linked to a suitably tagged asset, a persona-asset inheritance relationship will be shown.

CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

Search

test ▾

Home / Assets / Citrix Logs

SummaryCriticalityInterfaces

Asset

Citrix Logs

Shortcode

CL

Type

Information ▾

Description

Citrix log files

Significance

Accessible from the file system. Maybe contained sensitive information.

Tags

Persistence ✕

Enter new tags separated by comma

+ Environment

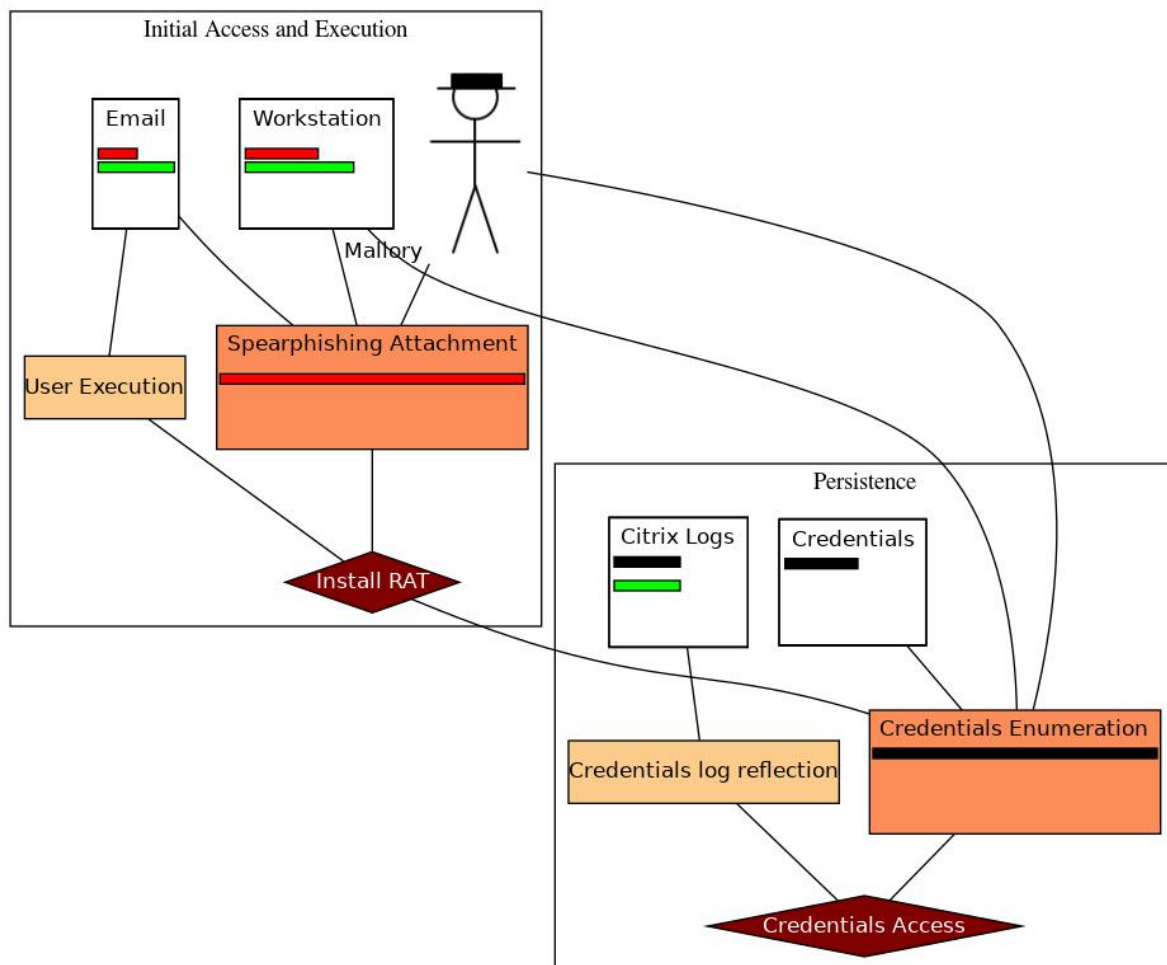
Default

DefinitionAssociations

+ Property	Value	Rationale
Confidentiality	Low	Requires authorised access to Citrix environment to view
Availability	Low	Necessary for application resilience.

Update

Cancel



CAIRIS

HomeSystem ▾Requirements ▾Risk ▾UX ▾Models ▾Options ▾

Search

Searchtest ▾

Home / Assets / Technician

Summary

Criticality

Interfaces

Asset

Technician

Shortcode

TECH

Type

People ▾

Description

Instrument technician

Significance

Intangible

Tags

role=Instrument Technician ✕

Enter new tags separated by comma

+ Environment

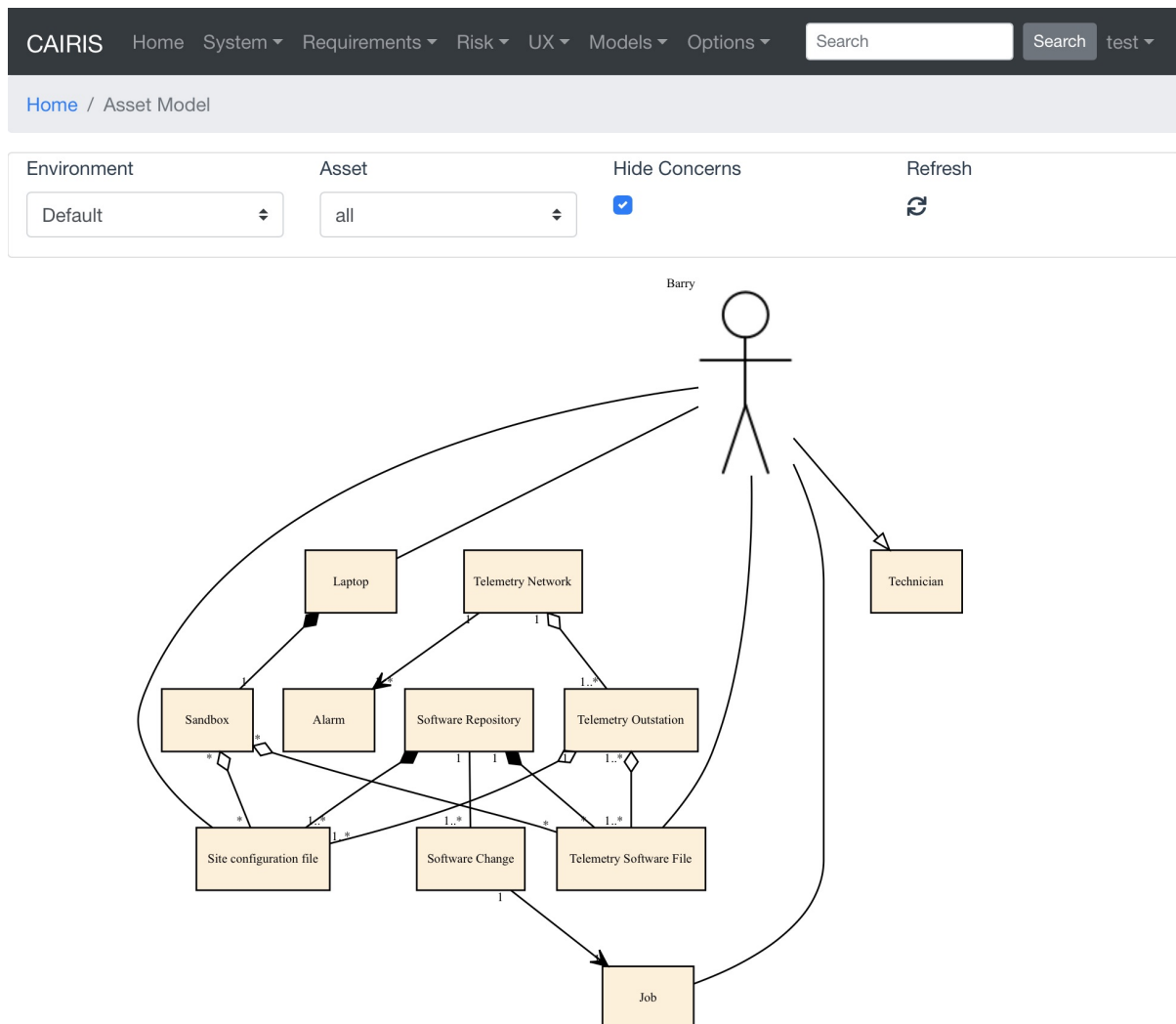
Default

DefinitionAssociations

+ Property	Value	Rationale
- Availability	High	Operations contingent on human staff.

Update

Cancel



Generating Documentation

The current contents of the CAIRIS database can be rendered as a requirements specification by selecting the System/Documentation menu. After choosing the type of specification, the output file name, and the output type – PDF, Word Document (.docx), OpenDocument Text (.odt), Rich Text Format (.rtf) – clicking on Generate will generate and download the specification document.

CAIRIS Home System ▾ Requirements ▾ Risk ▾ UX ▾ Models ▾ Options ▾ Search Search User ▾

Home / Export

Type ☒ Requirements ☐ Personas ☐ Data Protection ☐ Impact Assessment

File name

Format ☒ PDF ☐ Word Document (.docx) ☐ OpenDocument Text (.odt) ☐ Rich Text Format (.rtf)

CAIRIS currently supports the generation of 3 types of specification:

Template	Description
Requirements	A requirements specification that conforms to the Volere Requirements Specification Template
Personas	A specification document for personas.
Data Protection Impact Assessment	A DPIA specification that conforms with the ICO Data Protection Impact Assessments draft template .

37.1 Problems with wide models

Very wide goal or obstacle models can cause problems when generating PDFs, e.g. the webinos sample model. The workaround to such problems is to generate the documentation in Word, and export the model from Word to PDF.

37.2 Customising model files

Models in CAIRIS are rendered as SVG, and it can be useful to edit these models for improved readability. You can extract these models directly from the web app by installing the [SVG Crowbar](#) bookmarklet in your browser. The resulting SVG file can then be tweaked using an SVG editor like [Inkscape](#) , exported to the graphics format of your choice, and then added to your specification document.

CAIRIS server maintenance

If you have shell access to your CAIRIS host, there a number of scripts to aid in general server maintenance. These can be found in `cairis/cairis/bin` directory.

Calling these scripts with the `-help` flag will provide detailed information on the parameters they take.

38.1 Account management

You can add new accounts using the `add_cairis_user.py` script. Account names should be email addresses.

```
./add_cairis_user.py test@test.com test "Test user"
```

`cairis_users.py` provides a list of current users.

```
./cairis_users.py
```

`rm_cairis_user.py` can be used to remove accounts. All accounts will be removed if the parameter used is 'all'.

```
./rm_cairis_user.py test@test.com
```

The default database associated with CAIRIS accounts can sometimes get corrupted due to destructive operations (e.g. importing models) being interrupted. To re-create the default database for an account, you can use the `reset_cairis_user.py` script

```
./reset_cairis_user.py --reload 1 test@test.com
```

If you set the `reload` parameter 1 then CAIRIS will attempt to export the contents of the default database, and – once the default database has been re-created – attempt to re-import it. This can sometimes fail if the model contains reserved characters, but this can be overridden by setting the `ignore_validity` parameter to 1.

38.2 Importing and exporting models

Models can be imported using the `cimport.py` script. The below command, which is run from `cairis/cairis/bin`, imports the ACME Water sample model into the default database of the `test@test.com` user.

```
./cimport.py --user test@test.com --database default --type package --overwrite 1 .  
↪ ../../examples/exemplars/NeuroGrid.cairis
```

The *cexport.py* script can be used to export models.

```
./cexport.py --user test@test.com --database default --type package /tmp/NG.cairis
```

38.3 Backing up and restoring servers

backup_server.py creates a tarball containing exported model packages for all the default databases on a CAIRIS server, and a copy of the password hashes for each account.

```
./backup_server.py /tmp/backup140221.tar
```

If you have a clean CAIRIS server (i.e. with no accounts setup), you can use *restore_server.py* to recreate the accounts and account contents from a backup tarball.

```
./restore_server.py /tmp/backup140221.tar
```

Using the CAIRIS API

39.1 API documentation

API documentation can be found on [SwaggerHub](#).

SwaggerHub provides a virtual server at <https://virtserver.swaggerhub.com/failys/CAIRIS> you can use to quickly test get/post/put/delete methods on end-points without setting up CAIRIS, authenticating, etc.

```
curl https://virtserver.swaggerhub.com/failys/CAIRIS/1.0.9/api/attackers
[ {
  "theName" : "Carol",
  "theImage" : "Carol.jpg",
  "theDescription" : "Carol is a freelance journalist working in the South East of
↪England. Having heard stories about data theft, she is currently investigating
↪a number of e-Science projects, including NeuroGrid, to see if she can find a
↪story.",
  "theTags" : [ ],
  "theEnvironmentProperties" : [ {
    "theMotives" : [ "Headlines/press" ],
    "theRoles" : [ "Social Engineer" ],
    "theCapabilities" : [ {
      "name" : "Resources/Personnel and Time",
      "value" : "Medium"
    }, {
      "name" : "Resources/Funding",
      "value" : "Low"
    } ],
    "theEnvironmentName" : "Psychosis"
  } ]
} ]
```

```
}}
```

39.2 Authenticating with the CAIRIS server

For a more representative test, you'll want to run up cairisd and import a model. If you do this, the first thing you need to do is authentication to get a session. You can get this by posting to /api/session with your credentials. Let's assume our username and password is `test@test.com` and `test`:

```
curl -u test@test.com -X POST http://localhost:7071/api/session
Enter host password for user 'test@test.com':
{"message": "Session created", "session_id": "S9A3U7XkKEzqPwjwzKqR8jPGPVK0dvtf",
↪ "user": "test@test.com"}
```

By default, the session will point to the user's default database, but posting to `api/settings/database/{name}/open` can change the database the session points to, where name is the name of the database you want to point to.

When using the API in production use, the session should be included in header but, for development, you can add `session_id` as a parameter to the URL, e.g

```
curl http://localhost:7071/api/roles?session_id=S9A3U7XkKEzqPwjwzKqR8jPGPVK0dvtf
[{"theDescription": "Authorises access requests for NeuroGrid and responsible for_
↪ day-to-day administration.", "theName": "Certificate Authority", "theShortCode":
↪ "CA", "theType": "Stakeholder"}, {"theDescription": "Uses NeuroGrid data",
↪ "theName": "Data Consumer", "theShortCode": "DCON", "theType": "Stakeholder"}, {
↪ "theDescription": "Supplies data to NeuroGrid", "theName": "Data Provider",
↪ "theShortCode": "DPRO", "theType": "Stakeholder"}, {"theDescription": "Develops_
↪ NeuroGrid applications based on the provided NeuroGrid API and services.",
↪ "theName": "Developer", "theShortCode": "DEV", "theType": "Stakeholder"}, {
↪ "theDescription": "Professional or semi-professional hacker", "theName": "Hacker
↪ ", "theShortCode": "AKR", "theType": "Attacker"}, {"theDescription": "Uses and_
↪ supplies data to NeuroGrid", "theName": "Researcher", "theShortCode": "RCHR",
↪ "theType": "Stakeholder"}, {"theDescription": "Uses human frailty to access_
↪ computational resources.", "theName": "Social Engineer", "theShortCode": "SENG",
↪ "theType": "Stakeholder"}, {"theDescription": "Responsible for day-to-day_
↪ administration of NeuroGrid, including authorisation of access requests.",
↪ "theName": "Sysadmin", "theShortCode": "SYSADMIN", "theType": "Stakeholder"}]
```

39.3 The cairis_test database

As part of the quick setup process, a `cairis_test` database is created (password: `cairis_test`). Associated with this database is the `session_id` `test`. This database makes it possible to do general front-end development and testing without worrying about authentication.

You can import models directly into this database by using `cimport.py` without setting the user and database parameters. You can also use any end-points with this `session_id`, e.g.

```
curl http://localhost:7071/api/requirements?session_id=test
[
  {
    "theDescription": "Access to a NeuroGrid data-set shall be governed by an_
↪ access control policy.",
    "theFitCriterion": "None",
    "theLabel": "AC-1",
    "theName": "Dataset policy",
    "theOriginator": "Interview data",
    "thePriority": 1,
    "theRationale": "Need to determine which users can do what.",
    "theType": "Functional"
  },
  {
    "theDescription": "Requests for NeuroGrid access shall be authorised by the_
↪ nominated clinical exemplar sponsor.",
    "theFitCriterion": "None",
    "theLabel": "AC-2",
    "theName": "Access sponsor",
    "theOriginator": "Interview data",
    "thePriority": 1,
    "theRationale": "None",
```

(continues on next page)

(continued from previous page)

```
    "theType": "Operational"  
  }  
]
```


Over the years, CAIRIS has evolved to support new concepts and types of model. Its architecture has also evolved to make it easy for its [sadly] small development team to effectively maintain several hundred thousand lines of code. As a corollary, it is comparatively easy to extend CAIRIS, provided you follow the steps below.

40.1 1. Define the database tables

Each CAIRIS model is stored in its own MySQL database, so any new concept needs its own table or collection of tables. These tables need to be defined in `cairis/sql/init.sql`. This SQL script is called every time a new model is created, so it's important this script contains no errors. In many cases, errors occur if you forget to delete tables before creating them, or you define a table with foreign keys before defining its dependent data.

40.2 2. Define the database procedures

You need to create stored procedures for manipulating with your model data. These are defined in `cairis/sql/procs.sql`. As a rule, each model concept has stored procedures for (i) retrieving objects, (i) adding objects, (ii) updating objects, and (iii) deleting objects. As most objects are environment specific, there may be multiple procedures for (i) - (iii) depending on how complex the model object is. Take a look at some existing concepts like assets, attackers, and goals to see how these idioms are implemented.

40.3 3. Update the Python database proxy

`cairis/core/MySQLDatabaseProxy.py` is the module responsible for interacting with the model database, so you'll need to add methods for retrieving, adding, updating, and deleting objects. Again, looking at how this implemented using other CAIRIS should be a good source of inspiration.

40.4 4. Write your model object test case

Each model concept in CAIRIS should have its own test case in `cairis/test`. This effectively tests your stored procedures and methods in `cairis/core/MySQLDatabaseProxy.py` work correctly. The idiom used is to create test data in JSON, and to create a test case that retrieves, adds, updates and deletes model objects.

40.5 5. Update the CAIRIS DTDs

CAIRIS XML models are defined in DTDs within `cairis/config`. If your concept needs to go in a standard CAIRIS model file, it will need to be defined in `cairis_model.dtd`, but you may want to update other DTDs too. Because of how CAIRIS models are imported, the location of the concept in the DTD is important because you'll want to ensure any dependent objects are created first.

40.6 6. Update the model import / export code

To ensure your exported CAIRIS model contains your model object, you need to make a number of changes.

First, within `cairis/sql/procs.sql` are a collection of stored procedures for generating XML for model objects, e.g. `riskAnalysisToXml` for risk analysis related concepts. You need to edit the appropriate procedures to include the SQL necessary for retrieving your model objects and adding them to the generated XML. If you don't have to add a new stored procedure for your concept/s then this is all you need to do to ensure your exported model contains your new concept.

Second, CAIRIS uses SAX to parse model files during the import process. The different content handler classes used by the parser can be found in `cairis/mio`, and the appropriate class will need to be modified to create CAIRIS python objects to represent your model concepts. You will then need to update `cairis/mio/ModelImport.py` to ensure these objects are subsequently added to the CAIRIS database the model is being imported into.

Finally, depending on how fundamental your changes are, it might be sensible to also update the server-side `import` and `export` scripts too. These will provide you with a quick way of testing your import and export logic before delving too deeply into your API changes.

40.7 7. Implement the server end-points

At this stage, you can start thinking about implementing the code that will handle the API end-points. This involves updating and creating a number of files. First, you need to create a Data Access Object (DAO) objects for your model concept in `cairis/data`. In addition to acting as a wrapper for the database proxy, these objects are also responsible for marshalling Python objects to JSON (when retrieving objects), and vice-versa (when creating, updating, and deleting objects). Second, you need to define the object in `cairis/tools/ModelDefinitions.py` so Flask understands how to work this object. Third, you need to define the end-points themselves in `cairis/daemon/main/views.py`. Associated with each end-point will be an appropriate controller object in `cairis/controllers`. The object you choose will depend on the methods (i.e. `get`, `post`, `put`, `del`) you need to implement, and parameters you intend to use.

40.8 8. Write your API test case

At this point, you should add a test case to `cairis/test` to test your model API. If you look at other test cases, you'll see the norm is to import a CAIRIS model before kicking off your tests. To test the model import is working as it should, you might want to add your new model concepts to a CAIRIS model, import that, and try retrieving these in the tests. The other API tests should provide inspiration for how you should go about testing the different API end-points

40.9 9. Update the UI

Until now, all the changes made will have been to the [CAIRIS GitHub repository](#). However, to update the UI, you will need to update the code in the `cairis-ui` repository. Once the UI changes have been pushed to that repo, you should run the `installUI.sh` as described the [cairis-ui repository README](#).

40.10 10. Update the documentation generation process

`cairis/misc/DocumentBuilder.py` is responsible for interacting with the Python database proxy to rendering a Doc-Book specification, which forms the basis of generated documentation. This will need to update this to ensure your model objects appear in the specification. The module contains helper functions for generating things like lists and tables, so looking at how other model objects are handled should give you the knowledge necessary for incorporating your objects too.

41.1 Log files

The CAIRIS log files are a good place to look for signs of errors in the event of any problems.

If you are using the CAIRIS development server then the daemon will log directly to the console.

If you are using `mod_wsgi-express` then the log file will be saved to `/tmp/mod_wsgi-localhost:8000:0` (or similar depending on the port you've used).

If you are running Docker, you can get the latest entries and a running update of the log file with the following command:

```
docker exec -t `docker ps | grep shamalfaily/cairis | head -1 | cut -d ' ' -f 1`  
↪tail -f /tmp/mod_wsgi-localhost:8000:0/error_log
```

If you have setup CAIRIS to run as a system service then you can use `journalctl` to access the logs. For example, the command below will give you the latest log files and a running update.

```
journalctl -u cairis.service -f
```

For detailed logging information, change the `log_level` value in `cairis.cnf` to `debug`.

41.2 Raising issues

If you experience any problems using CAIRIS then please raise an issue in GitHub.

When raising an issue, please provide the version of CAIRIS you are using. You can find this by clicking on the System/About menu.

CHAPTER 42

Indices and tables

- `genindex`
- `modindex`
- `search`