Orchestration of Renewable Integrated Generation in Neighbourhoods

Grant Agreement Number: 314742

WP5 Deliverable 5.1: Energy Management System Validated in the Communities

---

WP5- Deployment and Validation of the Energy Control and Orchestration Algorithm

Delivery Date : August 2014

Update: March 2015

Dissemination Level : PU

Nature of deliverable : O

Deliverable Lead		
Organisation	**Contact**	**email**
University of Strathclyde (ESRU)	Paul Tuohy	paul.tuohy@strath.ac.uk

Contributors		
Organisation	**Community name (if applicable)**	**Contact**
ITI		Salva Santonja
Heriot-Watt University (MACS)		Manju Dissanayake
Heriot-Watt University		David Corne
ISA		Andreia Carreiro

1 Table of Contents

2 Scope and context for this deliverable

This deliverable requires the algorithms developed in WP4 to be deployed and validated on the hardware and software infrastructure established for each of the communities in WP3. As noted in WP3 the ORIGIN infrastructure as implemented has a 'cloud' basis i.e. a central server on which the ORIGIN software resides, rather than a server per community, so the deliverable is to deploy and validate the algorithms within the ORIGIN cloud system to service each of the communities.

This deliverable provides the platform to allow the ORIGIN system to run in background mode for 4 weeks during month 24 ahead of the ORIGIN system 'go live' in the Communities at the end of month 24 (D5.2). A report on the performance of the algorithms during this pre-release operation period will be delivered to coincide with the 'go live' date (D5.3).

The ORIGIN system actuation is to be phased and the first focus is on deployment of the ORIGIN system at community informational level. After the community informational level then building and system informational levels and system control actuations will be deployed in a phased manner. The phasing is to allow high focus on validation and the avoidance of risk.

Community meetings In April / May 2014 were held to scope out the user requirements for the system and these have been fed into the formats for the data outputs.

Two key elements are addressed in this deliverable, the first is the specification and implementation of the data schema to support the operation of the ORIGIN algorithms, the second is the translation of the algorithms into appropriate software including the scheduling script that calls them in the correct sequence. Finally, we also clarify how both the data schema and the implementation of the ORIGIN system become adapted to the specific buildings and conditions of the site in question.

3 Specification and Implementation of the ORIGIN Energy Management System data schema.

The initial ORIGIN data schema was established in WP3 to host the gathering of monitoring data. Some tables were included in this initial schema as placeholders for algorithm operation and user displays but these have now been revised in light of the algorithm development carried out in WP4.

The new data tables and parameters added to the schema in WP5 to support the community level algorithm operations are listed and described below, from section 3.2 onwards these cover the weather forecast, Generation forecast, Electrical load forecast, Green energy forecast (surplus or shortfall), and Performance metrics (historical performance).

3.1 Adaptation of data schema to Site Specifics

A consistent feature of our data schema design within ORIGIN is to ensure that the schema can be used with minimal adaptation for diverse installations (e.g. arbitrary new sites where the ORIGIN system may be installed). The schema itself therefore relies on no assumptions about the numbers and/or types of buildings monitored, and applies to sites where there is either wind generation, solar generation, or both. Although we have not done so for ORIGIN purposes, extensions of the schema to account for sites that have other forms of renewable generation would be simply accomplished.

Before providing detail concerning the data schema elements themselves, we indicate here some basic notes concerning the relationship between each element and site-specific issues:

- <u>weather forecast table</u>: the items included in this table are independent of any site specifics, and remain relevant to ORIGIN's functionality so long as external conditions are the key drivers of renewable generation at the site. The table focusses on the key weather items that are relevant to solar generation and wind generation.
- <u>Tables associated with renewables generation forecasts</u>: the current shape of the data schema assumes that streams of *actual* generation data (solar, wind or both) will be available at the installation site. This is an ideal basis for accurate forecasting, and it is the situation at the three ORIGIN demonstration sites. The data schema therefore assumes that there has been an installation phase (which we specify and recommend in various other deliverables concerning the ORIGIN process) in which community or other engineers have ensured that CT or similar sensors (see Deliverable 3.1) are in place to capture renewables generation data.
- <u>Tables associated with electrical load forecasts</u>: the ORIGIN setup phase, for any installation, involves (following community audit and questionnaires) the identification of (a) the community's national grid connection point(s), and (b) several individual homes and/or other buildings that will be monitored and/or partially controlled. In every case, a range of sensors (see Deliverable 3.1) will be installed. The electrical load forecast tables are set up to ensure that ORIGIN can handle the data appropriately whatever mixture of such load monitoring sensors are installed at the site.

- Table associated with green energy forecast: ORIGIN's "green energy forecast" implements a much-simplified analysis of the surplus/deficit profile (which, in detail, is used for optimisation and control) in order to provide a simple community level signal, available at community-level interfaces. The implementation of this in the data schema is generic and straightforward, enabling 'one-size-fits-all' for arbitrary communities. The values themselves adapt naturally to the community specifics (since they are calculated using the community-installed data-streams – see Deliverable 4.5).
- Tables associated with energy performance: in common with the data schema elements concerning electrical load and electrical load forecasts, the energy-performance parts of the data schema build on the ORIGIN setup and installation phase. During the latter activity, specific sensors will have been installed to capture data-streams from community renewables generation, community electricity load, and individual electricity loads for a variety of individual households and buildings in the community. The various derived (calculated) energy performance elements then depend directly on these data streams.

3.2 Weather Forecast

Table: weather_forecast

- weather_forecast_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- time_generated (timestamp with time zone, NOT NULL)
- wind_speed (real)
- wind_direction (real)
- temperature (real)
- solar_direct (real)
- solar_diffuse (real)
- cloud_cover (real)
- wind_speed_variance (real)
- temperature_variance (real)
- cloud_cover_variance (real)
- look_ahead (integer)

This table is refreshed once every hour. To read a complete forecast for a given site, a query of this table with site_id and time_generated will return 96 rows. The first row (with a look ahead of 1) will contain the forecast for time_generated + 30 min, the second row (with a look ahead of 2) will contain the forecast for time_generated + 60 min and so on.

3.3 Generation Forecast

There are two types of renewable electricity generation:

- Wind generation
- PV generation

For example at Findhorn, we'll be doing two predictions to cover these two types. This means two tables in the DB for the generation forecast:

- total_generation_forecast
- type_specific_generation_forecast

Table: total_generation_forecast

- total_generation_forecast_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- time_generated (timestamp with time zone, NOT NULL)
- look_ahead (integer, NOT NULL)
- energy (real, NOT NULL)
- energy_variance (real, NOT NULL)

Table: type_specific_generation_forecast

- type_specific_generation_forecast_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- time_generated (timestamp with time zone, NOT NULL)
- look_ahead (integer, NOT NULL)
- energy (real, NOT NULL)
- energy_variance (real, NOT NULL)
- generation_type (integer, NOT NULL) – Use 1 for wind generation and 2 for PV

Generation forecast, written to the “type_specific_generation_forecast” table first then “total_generation_forecast” table.

Example:

PV generation forecast created at 14:00 on 24 Sep for 15:00 on 24 Sep is 20 kWh

Wind generation forecast created at 14:00 on 24 Sep for 15:00 on 24 Sep is 75 kWh

In order to cover this, the type_specific_generation_forecast table contains the following values:

- site_id = 1, time_generated = 14:00, look_ahead = 2, energy = 20kWh, gen_type = 2
- site_id = 1, time_generated = 14:00, look_ahead = 2, energy = 75kWh, ge_type = 1

The total_generation_forecast table contains the following:

- site_id = 1, time_generated = 14:00, look_ahead = 2, energy = 95kWh

3.4 Electrical Load Forecast

Table: electrical_load_forecast

- electrical_forecast_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- time_generated (timestamp with time zone, NOT NULL)
- forecast_loadkwh (real)
- look_ahead (integer, NOT NULL)
- forecast_variance (real, NOT NULL)

3.5 24 Hour Green Energy Forecast

Table: green_energy_forecast

- green_energy_forecast_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- time_generated (timestamp with time zone, NOT NULL)
- look_ahead (integer, NOT NULL)
- green_energy_code (integer, NOT NULL)

The following codes will be used:

- 1 – display red
- 2 – display light red
- 3 – display light green
- 4 – display deep green

We write these codes to the DB rather than percentages. If we write percentages here, and if the green energy forecast then uses a set of rules to convert percentages to colours, then we will always have to change this set of rules to change the way the gadget displays colours. If we write a code, then the gadget will not have to interpret any rules, it can just go ahead and display the colour associated with that code.

3.6 Energy Performance Factors (EF)

These tables hold the performance metrics to be used to provide user feedbacks on current and historical performance. In general these hold historical values over some period of time – these values are either measured parameters or calculated parameters. The data tables established for the Community Informational Level are described here.

EF Value Tables

These are pre-calculated look-up tables defined for each country (i.e. GB, Portugal & Italy). The following tables are used for storing the pre-calculated values in the DB.

We start by adding the countries to the first table. Then we add months of the year to each country. We then add “week day” and “week end” for each month. Finally, we add week day and week end EF values. This approach is quite flexible. For example, if we decide to add Mon – Fri for one of the pilot sites, then we can do so without changing the structure of the tables.

Table: perf_met_country

- country_id (serial, NOT NULL)
- country_name (character varying(64), NOT NULL)

Table: perf_met_month

- month_id (serial, NOT NULL)
- month_number (integer, NOT NULL)
- country_id (integer, NOT NULL)

Table: perf_met_day_type

- day_type_id (serial, NOT NULL)
- day_type_name (character varying(64), NOT NULL)
- month_id (integer, NOT NULL)

Table: perf_met_time_slot

- time_slot_id (serial, NOT NULL)
- day_type_id (integer, NOT NULL)
- time_slot (integer, NOT NULL)

Table: perf_met_ef_values

- ef_value_id (serial, NOT NULL)
- time_slot_id (integer, NOT NULL)
- ef_value (real, NOT NULL)

EF Calculations Tables

Table: perf_met_so_far_today

The program runs once every 30 minutes and updates this table.

- perf_met_so_far_today_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- period_of_the_day (integer, NOT NULL)
- total_electricity_used (real, NOT NULL) (sum up relevant sensors for the last 30 minutes)
- renewable_generation (real, NOT NULL) (sum up relevant sensors for the last 30 minutes)
- grid_export (real, NOT NULL) (For Findhorn: IF E_DEM<E_W_GEN THEN E_EXP = E_W_GEN - E_DEM. IF E_DEM>E_W_GEN THEN E_EXP = 0.)
- grid_import (real, NOT NULL) (for Findhorn: IF E_DEM<E_W_GEN THEN E_IMP=0. IF E_DEM>E_W_GEN THEN E_IMP = E_DEM - E_W_GEN)

- renewable_generation_used (real, NOT NULL) (for Findhorn: IF E_DEM<E_W_GEN THEN E_W_USED = E_DEM. IF E_DEM>E_W_GEN THEN E_W_USED = E_W_GEN)
- site_emission_factor (real, NOT NULL) (for Findhorn: ((E_IMP*EF_GRID)+(E_W_USED*EF_WIND))/E_DEM)

Table: perf_met_yesterday

At the end of the day (at midnight), data stored in "perf_met_so_far_today" table is written to this table. This table will only ever have one row of data as this row will get over-written every day at midnight.

- perf_met_yesterday_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- electricity_used (real, NOT NULL)
- renewable_generation (real, NOT NULL)
- renewable_generation_used_locally (real, NOT NULL)
- proportion_of_ren_gen_used_locally (real, NOT NULL)
- emissions (real, NOT NULL)
- emissions_if_all_from_grid (real, NOT NULL)
- emissions_savings (real, NOT NULL)

Table: perf_met_this_month

At the end of the day (at midnight), data from "perf_met_so_far_today" table is added to this table. This table will only every have one row of data per pilot site. However, it will keep this line of data for a month. At the end of the month, the table will be cleared so that it can start holding data for the next month.

- perf_met_this_month_id (serial, NOT NULL)
- site_id (integer, NOT NULL)
- electricity_used (real, NOT NULL)
- renewable_generation (real, NOT NULL)
- renewable_generation_used_locally (real, NOT NULL)
- proportion_of_ren_gen_used_locally (real, NOT NULL)
- emissions (real, NOT NULL)
- emissions_if_all_from_grid (real, NOT NULL)
- emission_savings (real, NOT NULL)

Table: perf_met_since_launch

This table has one row of data per pilot site. At the end of the day (at midnight), data from "perf_met_so_far_today" table will be added to this table. This table will keep its data for as long as ORIGIN is live.

- perf_met_since_launch_id (serial, NOT NULL)

- site_id (integer, NOT NULL)
- electricity_used (real, NOT NULL)
- renewable_generation (real, NOT NULL)
- renewable_generation_used_locally (real, NOT NULL)
- proportion_of_ren_gen_used_locally (real, NOT NULL)
- emissions (real, NOT NULL)
- emissions_if_all_from_grid (real, NOT NULL)
- emission_savings (real, NOT NULL)

These tables have been established within the schema to enable ORIGIN Community Level Energy Management system operation. Similar tables are being established in support of building and system specific levels of system operation, these are being implemented and validated in support of the phased ORIGIN actuation plan and are not given explicitly here but can be found in the database schema.

4 Implementation and validation of the ORIGIN algorithms within the ORIGIN Energy Management System

As part of Work Package 4, the ORIGIN algorithms were partitioned into component parts. These include a weather prediction algorithm (WPA), renewable generation prediction algorithms (RPA), demand prediction algorithms (DPA), load shift opportunity prediction algorithms (OPA), and orchestration/control algorithms (OCA). Figure 2.1 shows the high level architecture of the ORIGIN algorithm including these sub-components.

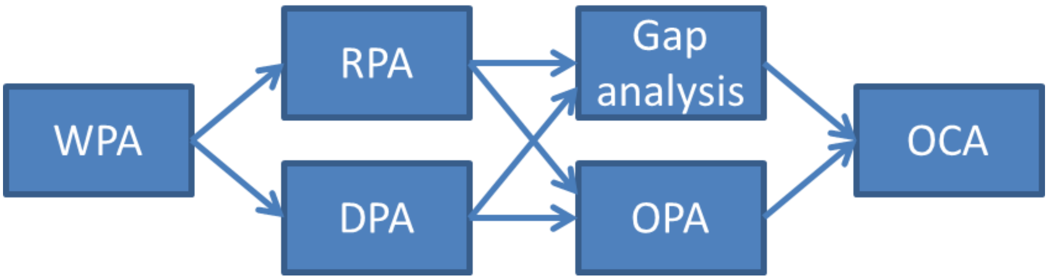


Figure 4.1. High Level Architecture of the ORIGIN algorithm (see above text for explanation).

Each of the algorithms developed in WP4 have been implemented in software within the ORIGIN Energy Management System in modular fashion. These algorithm modules have been implemented to be run with the Cron script and to operate together with the data schema described above to support the required ORIGIN functionality. Each module of the algorithms has been tested and validated against a range of data used in the algorithm development and validation phase, where possible gathered during the ORIGIN monitoring period.

In addition to the ORIGIN Energy Management System data schema the algorithm modules write out files which allow more detailed visibility of algorithm performance, these files have been used to generate records for algorithm validation purposes.

4.1 Notes on Adaptation to Site Specific Facilities and Conditions

In terms of implementation, we have tried to make the ORIGIN algorithms as generic and flexible as possible, so that adaptation to site specifics occurs naturally *via self-learning*, rather than via bespoke algorithm design. Nevertheless, a number of aspects of implementation are necessarily site-specific. The major ones are listed here:

- Parameterisation and configuration of demand forecasting: following community audit, setup and installation, a list of specific sensors are produced that specify where the ORIGIN system may find the data streams for the community itself, and for each individually monitored building. Input parameters for the demand forecasting algorithm are correspondingly configured to take account of this (for example, one command line input is the number of individual houses monitored). Finally, a routine is setup on the cron job that collects the latest data from these sensors every hour, and packages this as an input file for the demand forecasting algorithm.
- Parameterisation and configuration of renewables forecasting: following community audit, setup and installation, specific sensors are indicated that relate to (i) the site weather station, and (ii) site renewables generation. In the bootstrap phase of ORIGIN installation, a specific multivariate regression model is built, using these sensor streams (and other relevant data, depending on whether it is solar or wind renewables) to predict renewables for the site. This model is then integrated into the forecasting stage, and routines are set up to ensure it periodically re-learns its model from updated data.
- Parameterisation and configuration of load-shift opportunity prediction: following community audit, setup and installation, a further list of specific sensors are established that relate to devices that will be placed under ORIGIN's control. For example, for each water heater there will be a Temperature sensor, a Relative Humidity sensor, and a Cloogy device (to facilitate control signals). Prior to switching on 'control' within ORIGIN for such a device, a specific multivariate regression model is built, based on at least one weeks' monitoring of the device, using these sensor streams. For example, for a water boiler, this model learns the relationship between water temperature and external temperature, and the water/temperature time curves under different conditions. This model is then integrated into the opportunity prediction algorithm (OPA), and routines are set up to ensure it periodically re-learns its model from the updated sensor stream.

4.2 WeatherPrediction algorithm module

This is live. Everything has been scheduled using Cron and runs every hour.

On the Origin server, in addition to the data tables per the schema, historical weather forecasts for each timestep are available at:

- /home/origin_hw/Weather_Prediction/C_Weather_Forecast/Damanhur
- /home/origin_hw/Weather_Prediction/C_Weather_Forecats/Findhorn
- /home/origin_hw/Weather_Prediction/C_Weather_Forecast/Tamera

Forecast files are named like this: Tamera_wf_cloudCover_29_00_2014_1400.csv

Each file has 2 lines of 96 comma separated values. The first line has the weather parameter and the second line has the associated variances. For example, the first value in a file is the forecast for T + 30 minutes (half hour ahead forecast) and the second value is T + 60 minutes (one hour ahead forecast).

These weather forecast files have been used to validate that the implemented code gives the correct results against the same datasets used to validate during the algorithm development phase as described in D4.2.

4.3 Renewable generation Prediction algorithm module

Renewable generation forecasting is live. In addition to the data schema historical forecasts can be found in the ORIGIN server for each community eg. /home/ origin_hw/ Weather_Prediction/ D_Generation_Forecast/ Findhorn. The name of the file and the data format are similar to the format used in weather forecasting. We have trained and validated our models based on available training datasets.

4.4 Demand prediction algorithm module

Demand prediction based on simulation consumption data is live. Everything has been coded and validated using training datasets in such a way that when the actual site data becomes available, these modules can easily be switched from simulated data to actual data.

On the Origin server, the simulated consumption data is at: /home/ origin_hw/ Demand_Prediction/ Findhorn/A_Demand_Data. The demand forecast is at /home/ origin_hw/ Demand_Prediction/ Findhorn/ B_Demand_Predictions.

Demand forecast is in exactly the same format as the weather forecast (i.e. 2 lines of 96 comma separated values).

The implemented code replicates the performance which was established in the algorithm development phase.

4.5 Opportunities prediction algorithm module

The opportunity prediction module incorporates both a top down and a bottom up approach. The top down approach is based on identification of opportunities from statistical analysis of electrical demand profiles at the meter level and is championed by HWU, the bottom up approach championed by ESRU at UOS is based on the identification and characterisation of specific ‘firm’ opportunities associated with monitoring at the sub-meter level of thermal and electrical systems such as space heating, domestic hot water storage systems, electric vehicles, appliances.

4.5.1 Opportunity Prediction – Top Down Approach - Informational

Demand prediction provides the demand forecast for the whole community. However, we also work out the individual demand forecast for each metered house in the community. A number of "opportunities" can then be worked out for each metered house based on this. For example, if lots of demand is predicted at 17:30--18:00 for home 27, then we simply see this as an opportunity to move some of this to an earlier (e.g. 17:00--17:30) or a later (e.g. 19:00) time slot. The program finds around 10 to 12 such opportunities per home - they are (for the moment) based around peaks in the forecast demand for that home, and assume that they could be moved anything from 4 slots back to 4 slots forward.

The top down opportunities are documented in output files e.g. /home/ origin_hw/ Demand_Prediction/ Findhorn/ C_Profile_Opps

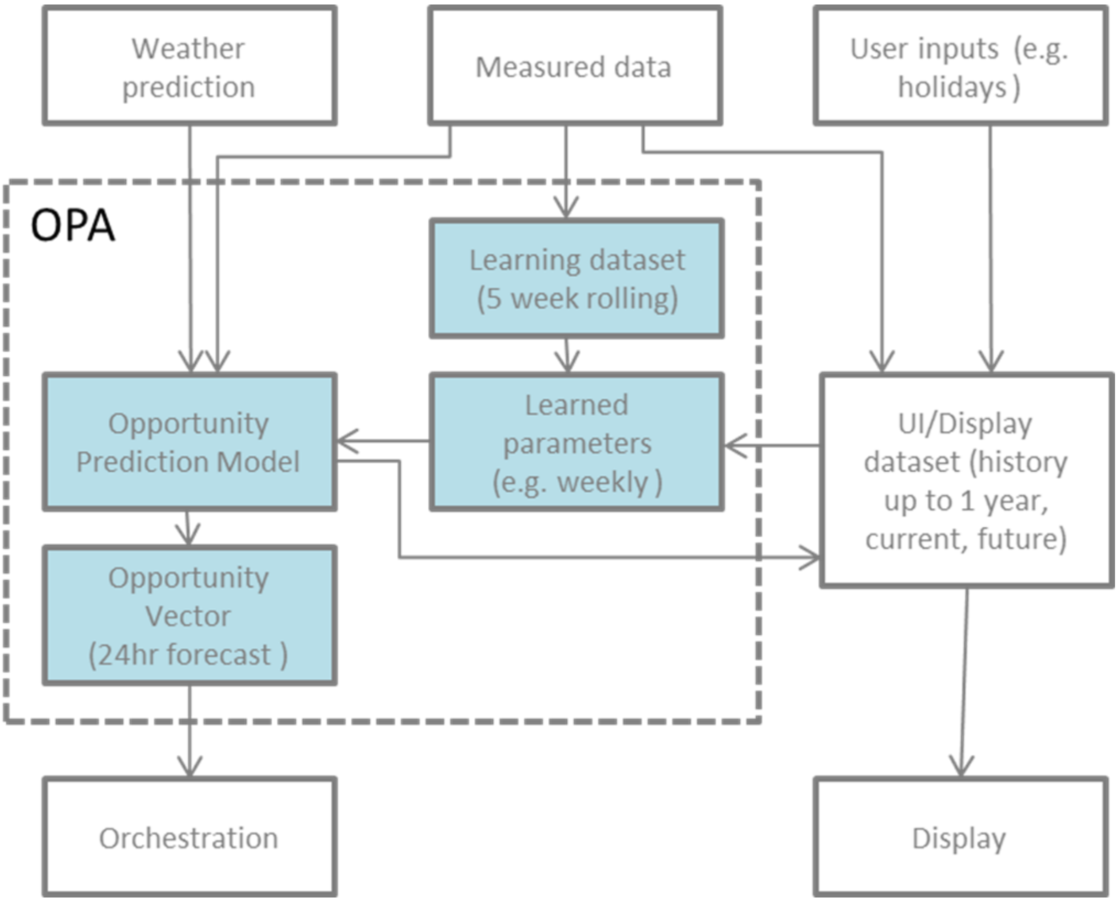
An example line in the opportunity file is: 5 0 0.469000 72 76 0

This translates to: for house ID 5, there is a type 0 opportunity to move 0.469 kWh from slot 72 to slot 76, at a cost of 0. Type 0 means this is a profile based opportunity and type 1 would mean an actuation.

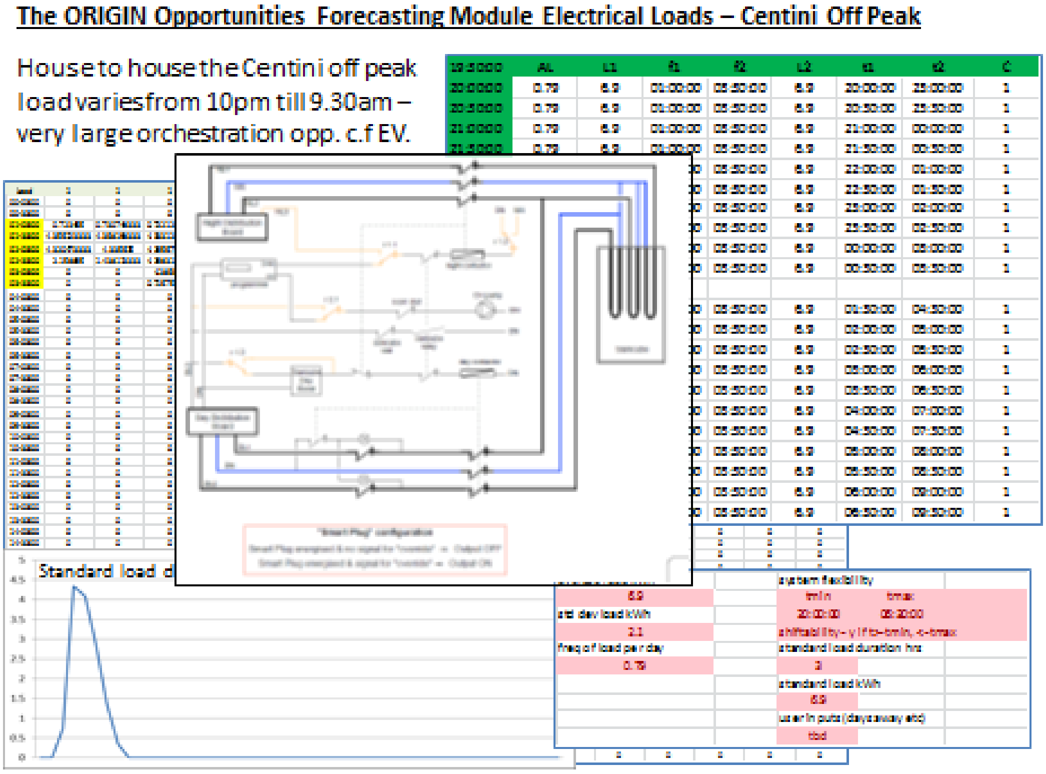
Please note that, for the moment, all the predicted opportunities are based on predicted demand for Findhorn which in turn is based on simulated consumption data. We have only considered "information only" opportunities for the moment. We are integrating actuation opportunities provided by the ESRU UOS team to this algorithm.

4.5.2 Opportunity Prediction – Bottom Up Approach – Informational and Control.

The algorithms developed for characterising opportunities based on monitoring at the sub-metering level as described in D4.3 and D4.5 have been translated into Java code and implemented on the ORIGIN server. The Java code has been validated as providing identical output to that of the algorithm development phase. Both the 'learning' and 'opportunity quantification' code have been implemented and validated, the separate elements of the opportunities prediction algorithm (OPA) are illustrated in the figure below.



Key steps in the bottom up approach are: 1. Get system parameters and flexibility, 2. Get 5 week dataset for weekly learning, 3.Run weekly learning algorithms, 4.Get weather data-feed per timestep, 5. Get system status per timestep, 6. Calculate opportunity using model, 7.Output opportunity vector. All elements have been validated and are ready to be deployed in the ORIGIN system in alignment with the phased implementation plan. The system is running in the background mode and files are being produced on the server for selected control opportunities which are to be the focus of the initial control actuations. The Centini Loads in Findhorn, the EV loads in Tamera and the water pump loads in Damanhur have been identified as having high potential. The software and hardware for Centini heating system load control is illustrated below.



4.6 Gap analysis and Orchestration/Control Algorithm

A working version of the code that produces an optimised set of load- shift actions to increase the communities' renewables untilisation (gap analysis and orchestration/ control algorithm)has been validated and integrated into the system and is live outputting files.

4.7 Cron job & Shell script

Everything has been scheduled using Cron and the shell script that glues everything together is working without issue at the moment.