

## **A practical example of a software factory: building a custom application for analysing EU Cyber Physical System (CPS) projects using Open Source software components.**

P. J. Palmer

The Wolfson School, ESoS Group,  
Loughborough University,  
Loughborough, UK.  
[p.j.palmer@lboro.ac.uk](mailto:p.j.palmer@lboro.ac.uk)

M. A. Sinclair

The Wolfson School, ESoS Group,  
Loughborough University,  
Loughborough, UK.  
[m.a.sinclair@lboro.ac.uk](mailto:m.a.sinclair@lboro.ac.uk)

C. E. Siemieniuch

The Wolfson School, ESoS Group,  
Loughborough University,  
Loughborough, UK.  
[c.e.siemieniuch@lboro.ac.uk](mailto:c.e.siemieniuch@lboro.ac.uk)

M. J. De Henshaw

The Wolfson School, ESoS Group,  
Loughborough University,  
Loughborough, UK.  
[m.j.d.henshaw@lboro.ac.uk](mailto:m.j.d.henshaw@lboro.ac.uk)

Copyright (c) 2015 P.J. Palmer. Published and used by INCOSE with permission.

**Abstract.** This paper is a retrospective analysis describing the development of a custom tool to organise data snippets derived from a substantial body of information, and a summary of the insights that this means of analysis provided in a very short time scale. The creation of data driven visualisations are of particular interest as they uncovered more cross-domain aspects of Cyber-Physical Systems projects than expert opinion had anticipated. These findings will be discussed fully in a second paper.

The focus here is the development of the "Vulture" data scavenging tool using Open Source software as system components to create a custom application to serve the data collection and analysis requirements of a REA (Rapid Evidence Assessment) work-package within an EU funded project, Road2CPS.

### **Background**

As the name indicates, the EU Funded Road2CPS project, a 2-year Co-ordination and Support Action within the EU-funded Horizon 2020 R&D programme, aims to create a road-map for the development, deployment and implementation of Cyber-Physical Systems (CPS) within the European community<sup>1</sup>. As one of its early outputs, the project used a collaborative approach to produce a comprehensive state-of-the-art report summarising the contributions of 53 EU-funded projects (current and recently completed) all concerned with the development of CPS and their ecosystems. The short time scale of six months to deliver this first report indicated that a REA-style approach based on written outputs from those projects would be a good tactical approach to the task at hand.

The five project partners included team members familiar with the CPS domain and able to make informed judgements about the sometimes specialist technical outputs from the projects. This

---

<sup>1</sup> <http://www.road2cps.eu>

presented an opportunity to use the available expertise to summarise the project outputs into short text snippets (usually about a paragraph), each categorised with meta-data (discussed later) to enable the production of the report within the time-scale and to develop data-driven visualisations. Given that the available expertise was split across four countries and five institutions it was necessary to contrive an approach that enabled the task to be broken down into a series of subtasks that could be completed by each of the contributors both independently and in parallel, with the ultimate goal of combining all the contributions into a coherent report.

For ease of reference, a guide to the nomenclature and abbreviations is given below for the reader.

### ***Nomenclature and Abbreviations***

**Cloud Web App.** In computing, a web application or web app is a client-server software application in which the client runs in a web browser. Google, Amazon and other providers offer cloud based hosting specifically for web apps.

**CPS.** Cyber Physical Systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. (Rajkumar et al. 2010, 731-736).

**Drupal.** Popular Open Source Content Management System (CMS) see <http://drupal.org>.

**LAMP.** LAMP is a web service stack, named as an acronym of the names of its original four open-source components: the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. The LAMP components are largely interchangeable and not limited to the original selection. LAMP is suitable for building dynamic web sites and web applications.

**Open Source.** Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.(Open Source Org 2015).

**REA.** A Rapid Evidence Assessment (REA) approach is useful for assessing research evidence on a particular topic, as comprehensively as possible, within the constraints of a given timetable. (Civil Service UK 2015).

**SIMILAR.** The acronym stands for State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. As their authors state, this process is quite “universal” and a considerable number of well known processes from diverse fields can be mapped to the SIMILAR process. Despite the linear appearance, the approach does not represent a sequential process. (Bahill and Gissing 1998, Ramos et al 2010)

**Software Factory.** A Software Factory is a development environment configured to support the rapid development of software applications. The concept pre-dates the capability to deliver such functionality. e.g. (Bratman 1975, 28-37).

**SoA.** State of the Art. In the context of this report, this was taken to be a description of the current limit of capability as inferred from the public domain information available on the 53 CPS projects assessed in this work.

## Adopting a Systems Approach

In this section we map the problem and its solution into the SIMILAR process. While a systems terminology was not used as the approach was applied, it was behind the mindset that led to the problem solution. Once again it should be emphasised that the approach is a parallel process, not sequential, and some commentary on the advantages gained are made within the following subsections.

### **SIMILAR**

**State the problem.** As described in the background section, the Road2CPS project was required to produce a comprehensive state-of-the-art report summarising the contributions of 53 EU-funded projects (current and recently completed) all concerned with the development of CPS and their ecosystems within six months of the start date. Table 1 presents the requirements in more detail. The constraints presented in Table 2 also did much to shape the final solution.

**Investigate alternatives.** The alternatives considered are presented in Table 3, A Morphological matrix of the subsystems considered. Early in the process the solution to the problem was seen primarily as a system of software and people due to the constraints of time and geography, and this shaped the morphological matrix. Under different constraints, perhaps high security and a team working in close physical proximity, solutions using dedicated hardware for servers may have been considered.

**Model the system.** A schematic of the functioning system is presented in Figure 1. To emphasise the non sequential nature of the system development, a functional demonstration was built and demonstrated in the first few days. The reasons for this were primarily political not technical: it was important to demonstrate that all team members could access the platform. Once all stake holders were confident that the system had the potential to deliver a functional solution the iterative development process could continue.

**Integrate.** Figure 3 also illustrates the subsystem integration. The use of stand alone packages loosely coupled to the core system has enabled many iterations and experiments to be performed to improve the tool.

**Assess performance.** The use of stand alone packages loosely coupled to the core system has enabled many iterations and experiments to be performed to improve the tool. For example, in the ambient Internet security and privacy have become an increasing concern. It has proved possible to upgrade the server to use encrypted SSL connections by default. Although this was never explicitly stated as a requirement it was a sensible enhancement that reduces the chance of a privacy issue. The following section explore the requirements capture of the Vulture Tool in more detail.

Table 1: Requirements for Custom Application

<b>Requirements</b>
Collate text and files.
Associate meta-data with text using terminology accepted by the CPS community.
Usable by all consortium members with minimal training.
Permit analysis of the data and collation into a report.
Control access to data using permissions
Be quick and simple to implement.
Updated as required.

Table 2: Freedoms and Constraints

<b>Freedoms</b>
The contributors were domain experts, so their opinions could be taken at face value.
A rich authoritative literature already existed with established classifications on which to base the meta-data.
Computer and internet access was easy for all contributors.
The working application was not a formal deliverable so need not have any working life beyond the immediate project.
Administrator access was available on desktop computer and LAMP server.
<b>Constraints</b>
Rapid development was essential as the primary task was to produce a report, not a software application.
The main contributors were distributed across five institutions in four countries. Face to face meeting time would need to be brief.
There was no financial or time budget set aside for the development of the application.

### **Deriving requirements for the vulture tool**

The requirements of the task were met by calling heavily on system engineering skills to decompose the requirements into a series of sub systems that were then realised with readily-available open source software components. The ad hoc assembly of these software system components as client and server software became the "Vulture Tool", a custom application with a single purpose for a single task on a single project. This systematised approach delivered the functional tool with a development time-scale of the key components measured in approximately 20 hours, ensuring effort remained focussed on the task at hand, instead of software application development.

This paper is a retrospective analysis of why the approach worked so well. Sufficient detail is presented here for this paper to provide a vade mecum for the approach to be applied to create other single use applications. As a cautionary note, the tidy narrative used within this paper describing the overall process does little to capture the almost "Scrapheap Challenge" atmosphere of assembling the application over a very short time-scale. The success undoubtedly depended upon the highly skilled contributions from multiple team members covering both CPS domain knowledge and

technical fluency in a range of programming languages to realise a working application along with the availability of the necessary platforms on which to run the application.

Finally, as the general purpose of the tool was to scavenge data, it was given the name "Vulture Tool" and a suitable Open-art image of a vulture modified and used as a logo.

### Defining the Approach

The overall approach of breaking the data into small snippets and the categorisation of those snippets was proposed and accepted by the partners at the initial project meeting along with a brief presentation of how the custom application would be assembled in the very short time-scale necessary. The key requirements for the tool were identified at this stage and are presented in Table1: applying a System Engineering viewpoint, the requirements of Table 1 were defined as four high level functional requirements: User Interface; Data storage and retrieval; Analysis; and Visualisation. Table 2 provides an additional refinement as it lists relevant freedoms and constraints that were considered alongside the requirements of Table 1.

The morphological matrix, Table 3, lists the software subsystems considered as candidates to fulfil these high level functions. The jump from the requirements of Table 1 to the functions and candidate solutions of Table 3 is a creative and abstract process that requires a working understanding across a wide range of technical topics. The morphological matrix (Ritchey 1998) is a useful way to concisely capture the essential elements of that process. The short time-scale dictated the need to use components that were known and available, rather than to research and find optimum solutions, so the components considered were filtered by an engineering bias of what seemed practical as a possible component. The availability of good quality Open Source software components complemented this style of working, as all the software trialled was zero cost, and had no direct financial burden on the project.

Working systems may be constructed by selecting one software subsystem from each row of Table 3. Even a simple matrix such as this offers 400 viable solutions, each with potential advantages and disadvantages.

Table 3: Morphological Matrix: Software Subsystems

Functional Requirement	Software Subsystem				
<b>User Interface</b> <i>Enter text, annotate with meta-data.</i>	<b>Use free text document (offline)</b> <i>Automated consolidation all but impossible.</i>	<b>Use Spreadsheet based form (offline)</b> <i>Requires multiple copies of the document in circulation. Management of updates once process started challenging.</i>	<b>Use Drupal CMS an on-line application interface (online)</b> <i>Requires admin access to LAMP server with FQDN. Quick and easy to install and set up. Single application served to all users supporting "on-the-fly" updates and development.</i>	<b>Google Spreadsheet based form (online)</b> <i>Collates data into a spreadsheet style database.</i>	<b>Cloud Web App hosting</b> <i>Django / python custom app hosting. Highly versatile but front loaded on effort.</i>
<b>Database</b> <i>Store and retrieve structured information.</i>	<b>Desktop client application: Base, Access, Spreadsheet etc (offline)</b> <i>Data must be imported from multiple contributors before analysis.</i>	<b>mySQL database (online)</b> <i>Interfaces well with Drupal. Easy to export data. Excellent support and migration to data analysis packages.</i>	<b>CSV file (offline)</b> <i>Static file format almost universally supported for input and output of data by database, spreadsheet and analysis applications.</i>	<b>Cloud SQL compatible data store (online)</b> <i>Used in conjunction with programmable web apps.</i>	
<b>Analysis</b> <i>Uncover useful interpretations of the structured data.</i>	<b>Spreadsheet (offline)</b> <i>Basic analysis easy. Difficult for more advanced analysis without scripting.</i>	<b>Drupal Views (online)</b> <i>Potential for on line data driven visualisations, using wyswyg interface.</i>	<b>R! (offline)</b> <i>Scripting language with many analysis libraries available as standard for manipulating data. Interfaces well with Ggobi data visualisation tool.</i>	<b>Python data analysis library (online / offline)</b> <i>Versatile custom analysis. Compatible with mySQL and other data stores.</i>	
<b>Visualisation</b> <i>Develop visually compelling representations of analysis outputs.</i>	<b>Spreadsheet (offline)</b> <i>Good support for simple visualisations</i>	<b>Google Charts (online)</b> <i>Good support for simple visualisations. Interfaces easily to Drupal.</i>	<b>Ggobi (offline)</b> <i>Tool supporting interactive data analysis and visualisation</i>	<b>d3.js - Data-Driven Documents (online)</b> <i>Visualisation tool especially suitable for HTML based applications.</i>	<b>Python data visualisation library (online /offline)</b> <i>Probably best suited for long term use due to the programming effort required.</i>

(Those used highlighted with grey background)

There are, of course, many other possible ways to deliver the functionality that are not considered here, some of which may also be good candidate solutions: an example would be macro scripts within an Office application suite. Although an excellent solution under some scenarios, macro scripts are quite correctly constrained by security settings and may not be executable by all parties involved. All solutions with an "obvious" deficiency such as this were simply dismissed to keep the development of the tool moving forward.

The following narrative illustrates the process of considering the essential freedoms and constraints of Table 2 in conjunction with the candidate solutions of Table 3, to describe how a suitable system was assembled from the list of potential components.

### ***Functional Requirement: User Interface functions***

Options:

- Free Text
- Spreadsheet
- **Drupal CMS**
- Google Spreadsheet
- Cloud App

Free text would be too labour intensive to analyse and would likely lead to a time overrun on the overall task. A spreadsheet might be a better choice, especially as macros and other custom functions may give a consistent output, but other strong negative features of spreadsheets described in the analysis section lead to the rejection of this choice. There would also be multiple copies in circulation so change management would also be difficult once the process had started. The Google Spreadsheet form would be relatively easy to set up and would be a single copy. However, offering users the potential to edit their contributions would not be possible, unlike Drupal where this is default functionality. The Drupal CMS offers a quick way of interfacing with a MySQL database and is easy to use in an on-going development mode. Space on a suitable LAMP server was also to hand. Cloud Web App hosting would be suitable if time and budget allowed, but the slow development necessitated by this approach led to rejection of this choice. Familiarity with the Drupal environment was an additional reason for its choice.

Drupal is an expandable CMS with a core system complemented by a library of optional modules that may be used supply additional capability. The Drupal installation for the Vulture Tool only took several hours to install along with a suitable set of modules configured to provide a suitable User Interface. Once the meta-data (see section below) had been defined and added as user selectable categories, data entry started in advance of the definition of the analysis and visualisation system components.

### ***Functional Requirement: Database functions***

Options:

- Desktop Client Application
- **MySQL**
- **CSV file**
- Cloud database

The selection of Drupal was predicated upon the use of MySQL but the other options are still relevant as SQL type databases offer a variety of migration paths to other database types. Desktop client applications are a realistic option, especially if they offer a route to suitable analysis and visualisation software. The CSV file format is almost universally supported as an input or output

format by analysis software. As a custom cloud app was not selected as the user interface, cloud SQL compatible data storage was not relevant.

Although MySQL was used as the online database, there were some practical reasons to use CSV as an offline format. The Analysis section describes why an offline application was chosen to fulfil the functional requirement. (See below). Drupal supports the use of a mySQL query builder called Views that enabled the export of the raw data as a CSV file. The CSV files generated by this process were well formed and suitable for import into the analysis software

### **Functional Requirement: Analysis functions**

Options:

- Spreadsheet
- Drupal Views
- **R!**
- Python libraries

While the data gathering process was under way, work started on evaluating analysis software using partial subsets of data. It quickly became clear that some on-line summaries would be useful, but that the real analysis would be undertaken offline. Drupal has many modules to facilitate integration with visualisation libraries, but it would be a mistake to consider these as a complete solution without careful consideration. The Python libraries may be used in both an online and offline context and they have the advantage of being robust mathematically and open to inspection. Using the Python libraries requires a considerable amount of programming, so they are not the best candidate for quick usage. However, they might be the best solution for a custom web based application, because of integration to MySQL and a web server. Similar comments apply to the JavaScript based libraries too, eliminating those from the list of candidate solutions.

Spreadsheets, while superficially attractive, are a poor choice for an in-depth analysis of complex data because of the well known issues with the built in statistical libraries and limited visualisation available (McCullough and Heiser 2008, 4570-4578; McCullough and Heiser 2008, 4570-4578;

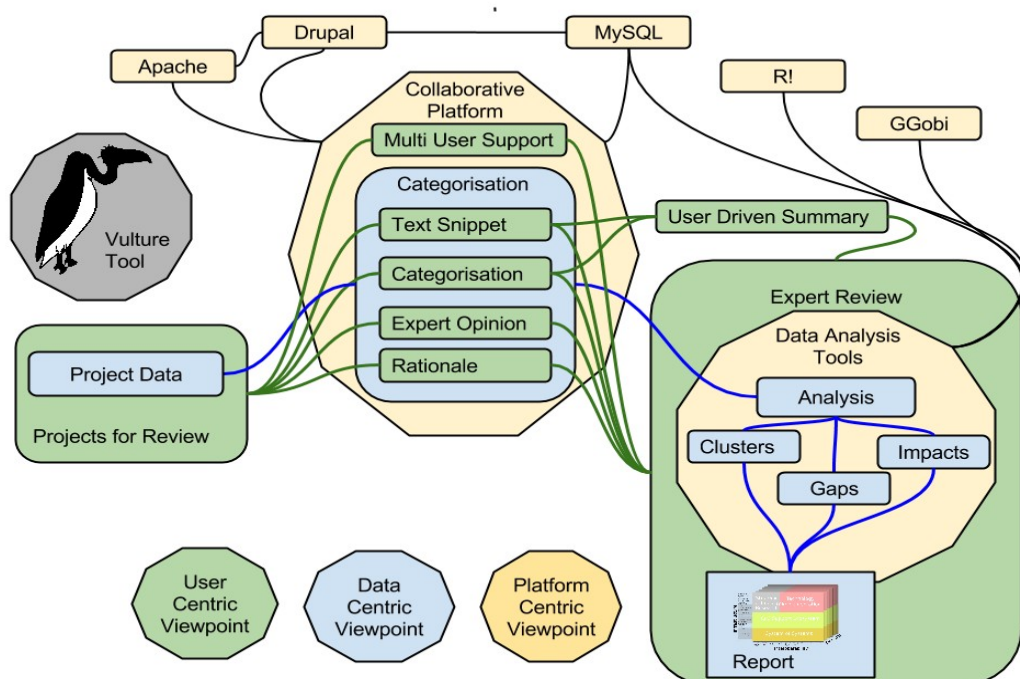


Figure 1 Schematic of Vulture Tool

Cryer, Should, and Marks 2001; Berger 2007, 2788-2791). Although these references are directed at Microsoft Excel, inspection will reveal many of the same limitations apply equally to other Spreadsheet programs as well. Often the only route forward for reliable analysis using spreadsheets is to add custom functions that can be inspected for accuracy, once again requiring custom coding.

The R! scripting language is mathematically robust and integrates easily with the GGobi data visualisation application (Cook and Swayne 2007), described below. The language is syntactically similar to Python and elegantly handles data sets as variables, making for concise, structured, easily readable code. A variety of helper development environments are available along with excellent documentation. Using a R! script-based approach enabled testing on incomplete data as work progressed, enabling debugging of the code prior to the formal analytical phase of the work. In retrospect, its versatility made it an excellent choice of scripting language to support the data analysis.

### ***Functional Requirement: Visualisation functions***

Options:

- Spreadsheets
- Google Charts
- **GGobi**
- D3.js Javascript
- Python visualisation libraries

Taking a few moments to consider the requirements of the task, even simple plots of data can be helpful. Many systematic errors are exposed on plots as out of range points and incorrectly shaped curves. Unexpected data points should always be queried, and if possible sample datasets may also help determine accuracy. The rapid development of the Vulture tool meant that ad hoc testing of the data as it was collected was essential, so some simple summary charts were created using Google charts and D3.JavaScript. However, these simple visualisations were not required for the final report, so never systematically validated and checked for accuracy. The limited visualisations of spreadsheets have already been discussed , and the necessary investment in time of using the python data visualisation libraries effectively eliminated them from consideration.

Again in retrospect, GGobi was an effective choice of a data visualisation tool. By primarily driving the analysis with scripts, the same analysis and visualisation could be applied repeatedly to different sub-sets of data. This enabled detailed checking of the analytic process to eliminate coding errors before committing to the final analysis for the report. As a further check, several stages of the analysis were saved as CSV files, to enable an audit of the intermediate steps.

A series of Drupal Views were used to output filtered subsets of data in CSV format. These CSV files were then imported into R! and GGobi. The same technique was used to migrate database text into word-processed documents as part of the final deliverable.

### ***Final Implementation***

The final schematic of the Vulture Tool is represented in Figure 1. In the practical realisation the Platform components were hosted as a sub-domain on a commercial LAMP server, and the Data Analysis Tools on an Ubuntu desktop (Ubuntu 2015). This was a practical split trading implicit requirements of security and convenience. It should be noted that technically there is no reason to split the LAMP server and Desktop platforms into separate hardware, as the required software subsystems can co-exist without conflict and the loading on the system may easily be handled by modern processors. However, ensuring a server is robust enough to be exposed to the internet is not a trivial task, so splitting the server and desktop platforms was desirable in terms of managing the security of the internet facing system components.



As a final refinement, the visualisations produced using GGobi were further developed by using the Inkscape vector graphics tool (Inkscape 2015) to produce publication-ready outputs, a process briefly described below.

## **Use of the Vulture Tool**

### ***Defining the Meta-data to use***

The 53 projects that were considered as part of this REA were confined to those funded within the EC (European Commission) Framework Programmes and in Horizon 2020. There are many other funded projects both within the EU and elsewhere around the world, but these were explicitly excluded a priori. In view of this, it was decided that the meta-data used would need to fit into a more generally accepted framework suitable for other CPS projects. The three dimensions of meta-data selected were:

- Domain of interest,
- Networking and Interoperability, and
- Infrastructure.

In view of the current and future significance of the industry-focussed ARTEMIS Joint Technology Initiative {{20 Gide, L. 2013;}}, its successor ECSEL, and the ARTEMIS Industry Association, it was decided that the Domain dimension should adopt the domains utilised in ARTEMIS:

- Environmental & agricultural information
- Healthcare
- Manufacturing
- Transport & Mobility
- IT&C
- Security
- Energy
- Smart Community

The Network & Interoperability dimension was developed from the generic Interoperability Framework (NCOIC 2006) and the energy-related Grid-Wise Architecture Council (<http://www.gridwiseac.org>), both concerned with interoperability within large-scale systems:

- Political/economic/regulatory/business board
- Business objectives: strategy and policy levels
- Business context: aligned operations
- Business context: aligned procedures
- Semantics: knowledge sharing
- Semantics: information sharing
- Syntactic interoperability
- Network interoperability
- Physical interoperability

The Infrastructure dimension was created partly based on ARTEMIS, for the same reasons as above, extended by prior knowledge within the Road2CPS consortium to include many more organisational aspects. This was justified on the grounds that the Road2CPS project is focussed on the steps to implementation, which necessarily will happen through organisations:

- Ubiquitous autonomy
- Architectures

- Big data
- Contracts & financial arrangements
- Resilience & fault tolerance
- Education
- Skills & training
- Human & machine awareness
- Interfaces & interoperability issues
- Methods/protocols/procedures
- Regulations & policies
- Standards & codes of practice
- Tools (including simulation)

It should be noted that any snippet of text could be annotated with multiple categories in each of the three dimensions. This increases the complexity of the analysis process and the manner in which it was addressed is described in the following sections.

### ***Interactions with the Vulture Tool***

To brief contributors on how to use the Vulture tool, a set of instructions was developed and two YouTube videos recorded to assist with consistency of input:

- Introduction to Vulture ([http://youtu.be/s7rF79A\\_Hol](http://youtu.be/s7rF79A_Hol))
- Creating a SoA Snippet (<http://youtu.be/Fi5tyAVUJv4>)

This level of help was successful as there were few queries from the expert contributors. Overall 255 snippets were collated and categorised, comprising: State of Art Snippets – 144; Gap Snippets – 59; Impact Snippets - 52

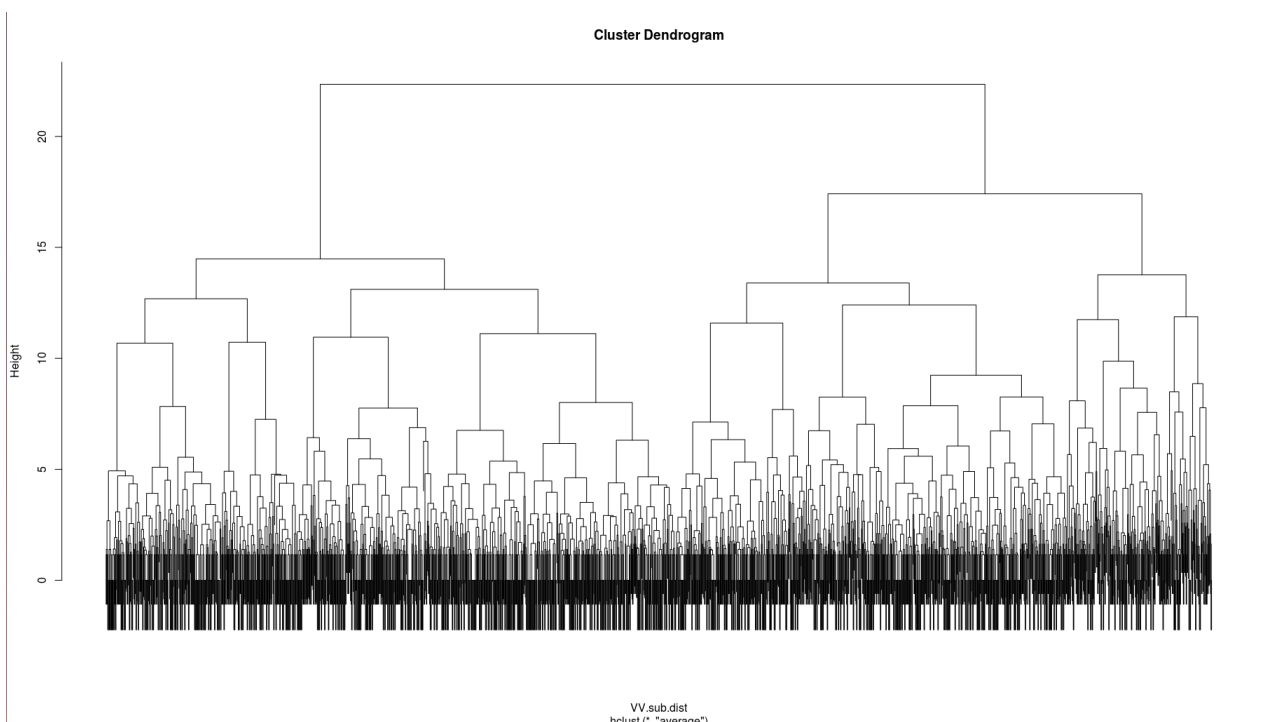


Figure 3 Clustering Process

## ***Analysis of the Output from the Tool***

A particular feature of the output from the Vulture tool is that the fields relating to the meta-data terms may comprise a list of multiple terms, rather than just a single term for each dimension. For compatibility with the analysis process, each field must be a single value, not a list of values. It was therefore necessary to "explode" the raw data into many rows of data, each with a single value for each field, prior to data analysis. The method chosen was a variant of "Bootstrap Re-sampling" (Efron 1979, 1-26).

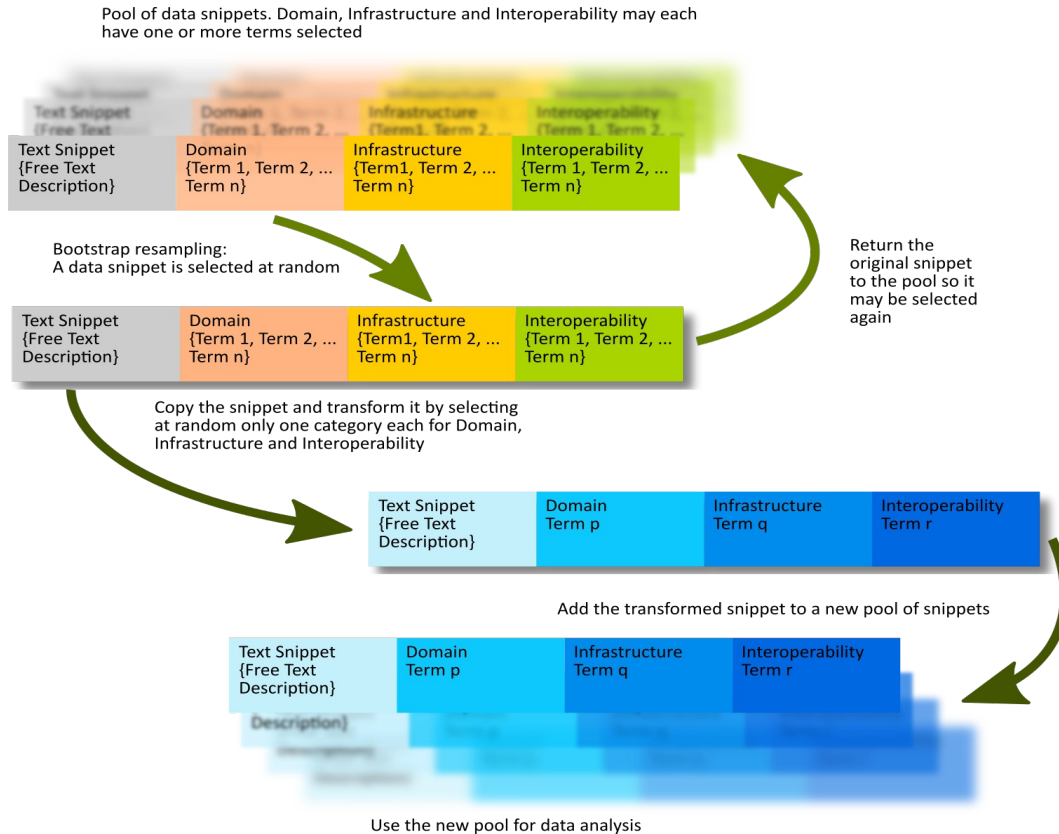


Figure 2 Schematic of Bootstrap Process

The process is represented schematically in Figure 2. A row of data is randomly selected from the initial pool of data. The selected row is transformed by randomly selecting a single term each from Domain, Infrastructure and Interoperability. This modified data row is now saved and added to a new pool of data. It is this transformed pool upon which the analysis is performed.

The bootstrap process is not sensitive to the number of meta-data categories applied to any particular data snippets, however, as the process was applied 10,000 times, the resultant data set was much larger than the original set. The efficiency of the R! programming language and the GGobi visualisation is such that there were no problems with data manipulation due to size. R! scripts were used to analyse the bootstrapped data and search for clusters using the inbuilt R! functions for hierarchical cluster analysis on a set of dissimilarities (ETH Zurich Department of Mathematics 2015).

The essential features of the overall approach from the REA perspective were:

- Contributor-introduced bias is unlikely as it was not possible to deduce the final visualisation during data entry;

- However, because of the scope of the CPS domain and its disciplines, there could have been bias introduced by the knowledge and background of the expert contributors. This was mitigated by the range and number of experts contributing;
- The final visualisations are underpinned by the overall dataset so may be verified by additional and subsequent analysis of the data.

The characterised CPS project data represents an additional output to the deliverable and may contain further useful information. The clusters have been derived by using a process that splits the overall data into a dendrogram or tree by effectively maximising the difference between members of the split as shown in Figure 3. The clusters are effectively generated by cutting the tree at the height where it splits into four. This process only works well on data with a reasonably balanced distribution, as indeed proved to be the case. Four clusters were found to be sufficient to obtain a useful characterisation of the projects. The definition of the four clusters using only the Interoperability and Infrastructure scales was, however, a surprising outcome, suggesting that the CPS projects reviewed have much broader application across domains than the project experts initially expected. This unexpected observation, underpinned by the data set and scripts relating to the analytical process illustrates the value of the overall data driven approach in helping to reveal useful information about the material being reviewed that might not have been otherwise uncovered. Further work will be required to confirm the observation and consider its potential impact on understanding the nature of the CPS projects reviewed; independently, the CPSoS project within the H2020 has arrived at the same conclusion.

The process of developing the graphic in Figure 4 which is a sample of how the findings were illustrated, is described here:

- The output directly taken from GGobi is a simple array of points. Each dot on the chart in Figure 3 below indicates that one or more snippets has meta-data that corresponds to that point. The cluster of points shown was retrospectively named Systems of Systems by examining the members of that cluster.
- Using Inkscape, multiple charts for each cluster are overlaid in separate layers and the coloured background groupings drawn to summarise those clusters. A stylised rectangular set of background shadings was chosen to avoid implying more precision than might exist in reality.
- Finally, to produce diagrams for the report, the axes are drawn in along with a representation of the Domain axis, which although it featured in the meta-data, did not produce any domain specific clusters. At any time a layer may be hidden or exposed to produce a series of consistent graphics for the reporting of the outputs.

## **Conclusions**

The approach used to create the Vulture Tool was successful in creating a custom application to assist a REA style review. It should be noted that the authors consider one of the primary reasons for success the availability of all the required expertise within the extended team.

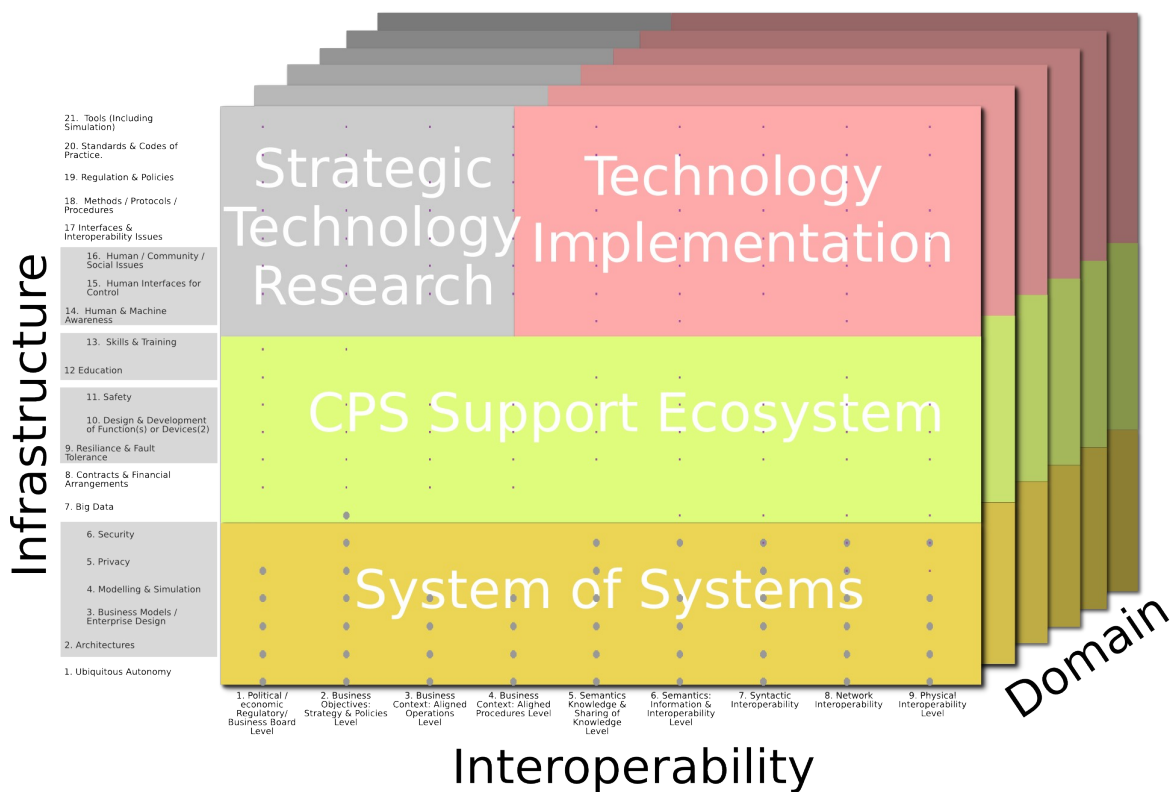


Figure 4 Output Graphics

A further positive contributing factor was the availability of a choice of software and platforms on which to run those applications as components, rather than being constrained to a purely managed corporate service. Focussing on Open Source software enabled the trialling of different candidate applications without cost penalty. It is interesting to note how the driver of speed and focus on final outputs lead to the particular configuration used in the Vulture Tool. Had the requirement been to produce a tool with a clear route to commercialisation, the final configuration would undoubtedly have been either:

- an installable stand alone application, or,
- a web based application accessed on a pay per use basis.

Neither of which would have been easily achievable within the tight time scale required.

Comparing the solution presented here to a hypothetical commercial application is of limited value since the ethos of this work was essentially for a single use application. However, software frameworks do exist that enable the rapid development and deployment of applications. The use of such frameworks is predicated upon access to the necessary expertise to make use of them. In this case no such expertise was available, nor was there time to learn.

While we do not suggest the systems approach described here is a panacea for all single project-applications, it has merits, and achieves goals in rapid application development first considered forty years ago when the concept of software factories was first proposed (Bratman 1975, 28-37). Where this approach differs from software factories and other platform based approaches is in the definition of the system boundaries. No assumptions were made that constrain the solution to sit within a particular hardware or software ecosystem; everything was regarded as a system component. This may be understood more clearly by considering the contrasting approach where

the system components are required to sit within a predefined managed framework or platform which may or may not have all the required functionality. Any changes to that framework will require debate and justification before implementation which may cause delays outside of the direct control of those trying to implement the system.

The technique could be replicated for other similar activities given access to the necessary skill set. Perhaps the most significant aspect of this approach is the ease with which fixes and upgrades could be made to the Vulture tool while data was being amassed. Of equal significance in this is that it never became necessary to make changes to the meta-data categories; this is a real benefit of prior knowledge, and indicates the importance of having experts available at the outset of such an exercise.

### **Further Work**

#### Further Work

It must be stressed that, due to the short time-scale, the focus of this work was on publicly available evidence, so it is quite possible that the final report understates the contribution the individual projects have made. Access to final reports was not possible due to their confidential nature and some projects may have been subject to a moratorium on technical outcomes to facilitate commercial exploitation. Further work is planned to clarify these potential shortcomings.

### **Acknowledgements**

The work described in this paper was funded under Horizon 2020 Coordination & Support Action: Road2CPS, “Strategic action for future CPS through roadmaps, impact multiplication and constituency building”. Grant number 644164.

The authors gratefully acknowledge the contributions made by the project partners from: Steinbeis Europa Zentrum, Germany; University of Newcastle, UK; Commissariat à l'énergie atomique et aux énergies alternatives, France; Anysolution SL, Spain; Atos España SA, Spain.

### **References**

- Bahill, A. T., and B. Gissing. 1998. Re-evaluating systems engineering concepts using systems thinking *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 28 (4): 516-527.
- Berger, Roger L. 2007. Nonstandard operator precedence in excel. *Computational Statistics & Data Analysis* 51 (6): 2788-91.
- Bratman, Harvey. 1975. The software factory. *Computer*(5): 28-37.
- Civil Service UK. Rapid evidence assessment toolkit index - civil service 2015 [cited 9/14/2015 2015]. Available from <http://www.civilservice.gov.uk/networks/gsr/resources-and-guidance/rapid-evidence-assessment> (accessed 9/14/2015).
- Cook, Dianne, and Deborah F. Swayne. 2007. *Interactive and dynamic graphics for data analysis with R! and GGobi*.
- Cryer, Jonathan D., Good Graphs Should, and Tick Marks. 2001. Problems with using microsoft excel for statistics. Paper presented at Joint Statistical Meetings, .
- Efron, B. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7 (1) (Jan.): 1-26, <http://www.jstor.org/stable/2958830>.
- ETH Zurich Department of Mathematics. R: Hierarchical clustering [cited 10/8/2015 2015]. Available from <https://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html> (accessed 10/8/2015).

- Gide, L. 2013. *Embedded/ cyber-physical systems ARTEMIS major challenges: 2014-2020*. ARTEMIS, DRAFT Addendum to the ARTEMIS-SRA 2011.
- Inkscape. Inkscape vector graphics editor. 2015 [cited 10/6/2015 2015]. Available from <https://inkscape.org/en/> (accessed 10/6/2015).
- McCullough, Bruce D., and David A. Heiser. 2008. On the accuracy of statistical procedures in microsoft excel 2007. *Computational Statistics & Data Analysis* 52 (10): 4570-8.
- Open Source Org. The open source definition | open source initiative 2015 [cited 9/22/2015 2015]. Available from <http://opensource.org/definition> (accessed 9/22/2015).
- Ramos, Ana Luísa, José Vasconcelos Ferreira, and Jaume Barceló. 2010; 2010. Revisiting the similar process to engineer the contemporary systems *Journal of Systems Science and Systems Engineering* 19 (3): 321-350.
- Rajkumar, Ragunathan Raj, Insup Lee, Lui Sha, and John Stankovic. 2010. Cyber-physical systems: The next computing revolution. Paper presented at Proceedings of the 47th Design Automation Conference, .
- Ritchey, Tom. 1998. General morphological analysis. Paper presented at 16th EURO Conference on Operational Analysis, .
- Ubuntu. Ubuntu: An OS for PC, tablet, phone and cloud. 2015 [cited 10/6/2015 2015]. Available from <http://www.ubuntu.com/> (accessed 10/6/2015).

## Biography

Paul J Palmer is a part time Research Associate at Loughborough University and Technical Director in an SME. He has interests in electronic design, manufacture, technology roadmapping and associated tools and methodologies. He has been involved in a number of systems orientated projects - including helping to commission the UK's first microprocessor-controlled nuclear reactor test and monitoring system for Plessey, and the manufacture of the control systems for UK's first generation of microprocessor controlled trains for the London Docklands Light Railway for GEC Traction. Paul is a Chartered Engineer and Member of the IET, a Senior Member of the IEEE and a Fellow of the Institute of Knowledge Transfer.



Michael Henshaw is Professor of Systems Engineering and leads the Engineering Systems of Systems (EsoS) Research Group. His research focuses on integration and management of complex socio-technical systems, with a particular emphasis on the challenges of through-life management of systems and capabilities. He joined British Aerospace (later BAE Systems) as an aerodynamicist and worked for seventeen years in aeronautical engineering tackling problems associated with unsteady aerodynamics (computational and experimental) and, later, multi-disciplinary integration. He was appointed to a chair in Systems Engineering at Loughborough in 2006 to direct the large multi-university, multi-disciplinary programme, NECTISE.



Carys Siemieniuch is a Professor of Enterprise Systems Engineering and is a member of the Engineering Systems of Systems Research Group in the Department. She has both UK professional and European CREE registration as an Chartered Ergonomist and Human Factors Specialist with expertise across the full range of systems-related human factors topics. Her key skills are in knowledge lifecycle management systems, organisational and cultural aspects of enterprise modelling techniques, organisational systems architectures, dynamic allocation of function and the design of complex systems. She is active in both the military and civilian domains.



Dr. Murray Sinclair is now a Visiting Fellow at Loughborough University. He is a Systems Ergonomist of some 40 years standing, having been an academic member of Loughborough University since 1970. His interests have evolved from the understanding of organisational processes of manufacturing from the shopfloor, through manufacturing systems engineering to design processes and the management of knowledge. Latterly, due to the steady infiltration of information technology into society and its pervasiveness in the lives of individuals, his interests now include the assurance of ethical behaviour by autonomous and semi-autonomous systems, such as robots, healthcare systems and the like. Murray is a Fellow of the Chartered Institute of Ergonomics and Human Factors.

