This is a repository copy of *A service oriented virtual environment for complex system analysis: Preliminary report*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/88699/

**Version:** Accepted Version

**Proceedings Paper:**

https://doi.org/10.1109/SYSOSE.2015.7151972

---

**Reuse**
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

---

White Rose
university consortium
Universities of Leeds, Sheffield & York

--

---

---
A service oriented virtual environment for complex system analysis: Preliminary Report

C E Dickerson, S J Clement
School of Electronic, Electrical and Systems Engineering
Loughborough University
Loughborough, UK.
{c.dickerson, s.j.clement}@lboro.ac.uk

D Webster, D McKee, J Xu
School of Computing
University of Leeds
Leeds, UK
{D.E.Webster, scdwm, J.Xu}@leeds.ac.uk

D Battersby
Jaguar Land Rover
Gaydon, UK
dbatters@jaguarlandrover.com

Abstract - Distributed virtual simulation is a capability that is increasing in demand within the automotive manufacturing industry. The distributed and networked approach to system level design and simulation stands to benefit from a unifying relational oriented modeling and simulation framework due to the large number of simulation technologies that must be integrated. This will also permit innovative use of existing independent simulations for increased concurrency in design and verification and validation. Through relational orientation, high level syntax and semantics for representing models and simulations have been developed for proof of concept analysis. This paper presents an approach to drive a process of analysis of the vehicle as a complex system through the combination of a relational trade-off analysis framework and a distributed simulation execution delivered through a service-oriented integration architecture. This promises to provide a rigorous, traceable and agile approach to early stage conceptual vehicle design and analysis.

Keywords: Design, V&V, Cyber-physical Systems, SOA.

1 Introduction

Original equipment manufacturers for automotive and aerospace vehicles are increasingly taking advantage of modeling and simulation (M&S) to reduce reliance on physical prototypes in the development life-cycle [1]. 'Virtual integration' supports design, simulation, verification and validation between environments; reducing the cost of testing through analyzing virtual solutions. The modern vehicle has become a complex cyber-physical system of systems requiring the integration of complex system and simulation models within its development process.

The ability to conduct a trade-off analysis for potential complex system solutions ideally would be supported by a closed, harmonized and holistic system model for analysis. However, in practice the required models are distributed amongst many pre-existing simulations. A common, formal and reusable framework for structuring design and analysis in such a distributed simulation environment has been lacking.

Individual components of the vehicle, whilst integrated at the physical level, are represented by domain specific simulations often created and governed by independent stakeholders. Therefore, a virtual integration approach must consider the combination of system level behaviors and a distributed systems view of these disparate domain simulations. Understanding the process of vehicle design and verification over a distributed simulation network in a dependable way demands substantial advances in how design models and simulations are modeled compared to the more commonly used approach of tightly integrating simulations on a local execution environment [2,3].

Our proposed methods are illustrated through an elementary case study. We demonstrate how a relational representation of a vehicle transient drive cycle can be utilised to prepare for integrated simulation in a distributed network of individual simulators; orchestrated through a service-oriented analysis workflow of integrated simulations.

The remaining structure of this paper is as follows: Section 2 presents an overview of the challenges for complex system M&S. Section 3 describes our proposed M&S framework for complex systems. Section 4 provides a case study to apply our approach to modeling and simulating the effects of driver behavior on vehicle performance. Section 5 outlines the conclusions and future direction of this ongoing work.

2 M&S for Complex Systems

Design specification in traditional engineering practice uses various methods to specify system elements (components/subsystems). Properties of each element are specified, e.g. by an attribute value and a tolerance on that value. Aggregating these specifications to system level attributes and functions is not always clear in current practice. Relational orientation has been developed to provide a more natural approach to such aggregation and system integration.

In complex systems (and systems of systems), system level analytics typically do not exist; therefore sub-systems are simulated individually. Relational orientation can be especially useful when designing and simulating systems or systems of systems for which there are no reliable and repeatable overarching system analytics.
In order to simulate dynamic system responses, analytics must be executed using their defined mathematical functions in the order in which the system performs its functions. In the simulation process, these (static) analytics must become an executable used for analysis of system response to dynamic change. Therefore in Section 3, time will be introduced along with system architectural elements to include control elements defined and integrated into the system specification.

3 M&S Framework

The M&S framework will be implemented using a Relational-Oriented Systems Engineering and Technology Trade-off Analysis (ROSETTA) framework [5]. This permits translation between mathematical models, analysis of physical systems, and disparate computer simulations. It provides a unified common framework for both design and V&V; filling the gap at the top of the systems engineering V-model and capturing the relationships between system input variables and system objectives or requirements. While similar to the Quality Function Deployment (QFD) House of Quality, ROSETTA replaces expert opinion with mathematical relations. Aerospace and data link applications of ROSETTA are presented in [5], [6].

3.1 ROSETTA for complex system of systems

The central concept is to use available models of the system or its components, e.g. mathematical, simulation or data models to create a static relational structure of design solution space in which the time dependency is not exposed. If a system level model is not available or achievable then lower level models can be used to create the pairwise sensitivities between the attributes of the operating environment and those of the system.

Figure 1 shows an abstract view of a ROSETTA framework. After first identifying the input variables and objectives of the stated problem the static relational structure can defined. The Q matrix is defined first, capturing the relationships between the input variables and the objectives. These could be sensitivities (partial derivatives) of transfer or response surface functions. If there is no coupling between input variables or objective variables, then the transformation matrix Q alone provides the static relational structure. These transformation relationships are sufficient for design and dynamic simulation.

Any coupling between the objective variables is stored in the M matrix and coupling of the system variables are stored in N. The collective matrices M, N and Q together define the static framework.

In the general problem, the partial derivatives at a given point in the design solution space, or estimates of their values can used to populate the Jacobian matrices of the transformational matrix and of the system matrices. When properly combined using the chain rule, the resulting total differentials give system level directions of improvement for the design variables. These are used in ROSETTA in place of a system level model or analytic when none is available.

3.2 Using ROSETTA for system simulation

When no system level model or analytic is available due to the complexity of the system or system of systems of interest, the process of developing a relational oriented framework for a simulation workflow is shown in Figure 1. To illustrate this, a static framework will be developed in Section 4.2 and extended in Section 4.3 to a dynamic structure by appending time as a parametric to the system matrix in the modeling and analysis case study.

The result will then be a partial differential equation for the total derivatives of the objective variables with respect to time, in which the stable relations are captured in the matrix structure of the framework.

Thus, the key for provision of a unified common framework for both design and V&V is to create a ROSETTA framework of the (static) relational structure of design solution space to which time differentials can be appended for dynamic simulation of candidate solutions. This will be a subject of the case study in Section 4.

3.3 Integration of Distributed Simulation into the M&S Framework

As described earlier in the paper, in order to apply the M&S framework to a production engineering environment, it is not possible to assume that the high fidelity domain simulations and subsystem level analytics are contained within a closed execution environment. In practice, these systems will be (physically) distributed across an organization and often developed in independent stovepipes [7]. Integration of these simulations with the M&S framework requires not just network enablement, but also the harmonization of heterogeneous interface specifications and modeling assumptions. The development of domain specific simulations has proven successful in the automotive sector, however, the networked and distributed integration of these domain simulations still remains a challenge.
of each individual service is transparent to the engineer allowing them to focus on achieving the most accurate results as fast as possible. Based upon semantic models of the services, the VIDAE is able to check the compatibility of connected services in addition to alerting the engineer whenever manual intervention is required to resolve mismatch between service inputs and outputs.

Once a workflow is configured, including appropriate analysis services along with design constraints, the engineer is able to observe the results and sensitivities of the designed vehicle system. Figure 3 provides a high-level overview of the Analysis Workflow consuming an abstract model and workflow specification through to generating design results. The system is comprised of two major subsystems:

- The System Analysis Framework analyses each individual service within the workflow. Firstly it is verified whether the services compatible with each other given various domain models as well QoS constraints. Then System Design Framework is utilized to identify the parameter sensitivities (using ROSETTA). These results are then integrated to provide a workflow analysis.

- The System Design Framework seeks to find the most optimum design by minimizing the sensitivity of the entire workflow. The results from running simulations through DIVIDER are integrated and analyzed using the methods described in this paper.

### 3.3.3 Distributed Virtual Integrated Development

To support the distributed Analysis Workflow we developed the Distributed Virtual Integrated Development Environment (DIVIDER) [12]. This provides a Service-Oriented environment for integrating domain simulations dependably and in real-time. This is underpinned by a powerful workflow technology that adapts to changes in the execution environments to satisfy the QoS requirement for the workflow. In cases where delays are encountered, simulation response time is prioritized over fidelity and lower fidelity simulations can be utilized.

SOA abstraction of the capability of domain simulations from their implementations permits the agile combination of services by the engineer and the rapid interchange between implementations based on a semantic model of their functionality. Figure 3 outlines the architecture of the DIVIDER utilizing the entire SOA system stack from execution platforms through to the abstract workflow. DIVIDER breaks the concept of a workflow engine into three logical components which different phases of execution: offline, deployment, and online. The concept of a publisher is also introduced to automate the process of service publication. Finally the data schema provides the mechanism by which incompatible services can potentially be integrated.
4 Case Study: Simulating Effects of Driver Behaviour

The automotive domain provides a good example of a system domain that has high fidelity subsystem level analytics, simulations or test data models but no reliable and repeatable overarching system level analytic or simulation. There is no single analytic to support important design trades such as optimizing system design for a key performance parameter, e.g., fuel consumption constrained by regulatory requirements on emissions and CO2. The aim of this section is to demonstrate early research results of how ROSETTA and a Service Oriented Virtual Environment can be used to meet this challenge.

4.1 Analysis in a driving course transient cycle test

Governments and agencies have specified extensive tests using drive cycles to assess whether vehicle emissions and CO2 satisfy regulatory requirements [13]. Driving cycles are generally defined in terms of vehicle speed and gear selection as a function of time. Speed profiles consist of $n$ data rows of time in seconds $t_i$ ($1 < i < n$) and speed $v_i$ in km/h ($1 < i < n$). The drive cycle can be performed in either a full-vehicle test or on a rolling road. Figure 4 provides a stylized sample of an EU drive cycle in graphical form.

The drive cycle in Figure 4 is a section of a transient type. Drive cycles can be broadly divided into ‘steady state’ and ‘transient’ drive cycles.

- A steady state cycle is a sequence of constant engine speed and load modes. These are not the focus of NEDC cycles for light-duty vehicle models.
- A transient cycle is a sequence of constant accelerations, decelerations, and speeds in the vehicle speed and engine load are more or less constantly changing.

Driver behavior will affect the level of emissions. The simplest example is the actual accelerations realized in a real or simulated test. This is illustrated in the drive cycle illustrated in Figure 4.

4.2 A ROSETTA framework for a driving course transient cycle test

The goal of this section is to specify the mathematical models that govern the dynamic behavior of fuel consumption and emissions. An elementary ROSETTA framework will be developed to structure these as a model of the objectives, a model of the vehicle, and a transformation model between the two. Time integration through the drive cycle can then be accomplished by making calls to simulations or databases as the vehicle traverses the time-velocity waypoints of the test.

Three objective variables have been identified for the emissions problem case study. Fuel consumption is sought to be minimized subject to constraints on emissions. For the purpose of illustration these will be limited to carbon monoxide (CO) and nitrogen oxide (NOx). Let $z_2$ and $z_3$ be the mass of CO and NOx emitted during the complete drive cycle measured in grams. These must meet regulatory constraints, which are specified in g/km. The total fuel consumed is denoted as $z_1$. This is nominally measured in liters but for the purpose of analysis is specified in kg. The fuel economy $z_0$ is commonly represented as the derived quantity $z_0 = \frac{\text{kilometer}}{\text{liter}}$ where $s$ is the total distance travelled in the test and $\kappa$ is the conversion between kg and liters of the fuel. The key performance parameters of the vehicle are acceleration (m/s$^2$) $y_1$ and speed (m/s) $y_2$.

The Jacobian matrix in the central section of Figure 5 specifies all possible sensitivities of the objective variables to vehicle variables. For specified test conditions, each of these partial derivatives can be assigned numerical values that can be stored in an array. The emissions variables $z_2$ (CO) and $z_3$ (NOx) are the result of imperfect combustion and can be regarded as mass fractions of the amount fuel consumed ($z_1$). The lower left section of Figure 5 depicts the sensitivity of CO and NOx to changes in fuel consumed. These sensitivities are typically derived from large databases collected from bench test measurements of an engine under specified load and other conditions. In the lower left of Figure 5, the $M$ matrix is a reduced Jacobian matrix in which the symmetric partial derivatives (i.e. the partials of $z_1$ by $z_2$ and $z_3$) and the negligible or zero derivatives have been ignored. These two couplings will be the only ones considered in the objectives model.

There is one coupling to consider in the relational structure for the vehicle. This is between the vehicle speed and acceleration. Specifically, the relation $y_2 = ty_1$ (i.e. speed is acceleration times time) yields the sensitivity $t$ of $y_2$ to $y_1$.

The coupling of $y_2$ and $y_1$ exposes an explicit time dependency of these vehicle variables to time. This permits augmenting the structure with time, as indicated by appending the $2 \times 2$ matrix with an exterior row and column for time. As such, the new $3 \times 3$ matrix is not intended to represent three vehicles variables that may have coupling but rather two that are defined parametrically by time, i.e., $y_1 = y_1(t)$ and $y_2 = y_2(t)$.
the time derivative of the case study because the acceleration is constant. Further, the objectives during a drive cycle are given by:

Furthermore, time has been properly factored out of the differentials in terms of the partial differentials.

This now makes clear how to express the time derivatives of acceleration as zero in Figure 5, equation (1) as a third variable in the system model. Instead, $t$ is the parameter through which the system variables are defined dynamically. For the objectives $z_{1-3}$, using the time derivative of acceleration as zero in Figure 5, equation (1) for the drive cycle simplifies to:

$$\frac{dz_1}{dt} = y_1 \frac{\partial z_1}{\partial y_2}$$

(3)

Recall that when $i = 1$, $z_i$ is the fuel consumed. The partials with respect to $z_i$ are just factors of 1 and drop out of the equation. For emissions $z_2$ and $z_3$, the equation picks up an additional factor $\frac{\partial z_{2,3}}{\partial z_i}$ that accounts for the mass fraction of fuel that is converted to an emission. Other than this factor, simulating emissions is the same as fuel consumption. Each factor in equations (2) and (3) can be computed by independent simulations (e.g., a driving profile, mileage model and emission simulation).

This equation supports dynamic simulation by replacing $dt$ with a time increment $\Delta t$. The right hand side is constant through the time increment. For the case of a drive cycle with acceleration, the product of the time increment with acceleration and change of fuel consumption with respect to speed yields a non-zero increase to the rate of change of fuel consumption. For the case of a cruise cycle, the acceleration $y_1$ is zero and the whole right hand side vanishes. The fuel consumption then remains constant over the cycle.

### 4.4 Specification of Analysis Workflow

The purpose of simulation and analysis in the emissions case study is to provide objective evidence for the evaluation of system level behavior and performance in relation to the intended design performance. The equations of the previous subsection are not system level analytics where design solutions are given by the assignments of values to the variables. In fact, due to their differential form, these equations are suited for local rather than global analysis of the design solution space. Nonetheless, the equations can be used for simulation of system level performance in the neighborhood of specific design solutions.

The distinction between the workflow based on ROSETTA and customary discrete event simulation is that the coupling of variables both in the objective and in the system model can be accounted for when the system simulation is distributed across a number of independent simulations. The verification of the workflow and application to conceptual analysis using response surfaces permits replacement of the differential operations in the cells of the ROSETTA framework with purely algebraic expressions that admit numerical calculation. The numerical values in the cells of the framework will depend on the state of the system to the extent that there is coupling. In the case of linear responses the partial derivatives in the transformation matrix are simply the coefficients of the linear expressions and these do not change with system state.

### 4.5 Making service calls

The implementation of the case study will be concerned with the provision of the computational workflow to a distrusted service oriented simulation environment. The

<table>
<thead>
<tr>
<th>$*$</th>
<th>$*$</th>
<th>$*$</th>
<th>$*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>$y_2$</td>
<td>$t$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\partial y_1}{\partial t}$</td>
<td>$\frac{\partial y_1}{\partial y_1}$</td>
<td>$\frac{\partial y_2}{\partial y_2}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\partial y_2}{\partial y_1}$</td>
<td>$\frac{\partial y_2}{\partial y_1}$</td>
<td>$\frac{\partial y_2}{\partial y_2}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\partial z_1}{\partial z_1}$</td>
<td>$\frac{\partial z_1}{\partial z_1}$</td>
<td>$\frac{\partial z_1}{\partial z_1}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\partial z_2}{\partial z_1}$</td>
<td>$\frac{\partial z_2}{\partial z_1}$</td>
<td>$\frac{\partial z_2}{\partial z_1}$</td>
<td></td>
</tr>
<tr>
<td>$\frac{\partial z_3}{\partial z_1}$</td>
<td>$\frac{\partial z_3}{\partial z_1}$</td>
<td>$\frac{\partial z_3}{\partial z_1}$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. ROSETTA framework for simulation
assignment of a numerical value to each variable and partial derivative of the equations in the previous section becomes a service call to a simulation. For the fuel flow calculation, we envision there would be two service calls. The assignment of a value of acceleration to $y_1$ for the simulation of a profile is a service call to a driver behavior model. The assignment of a value of the sensitivity of the fuel consumed ($z_1$) to the vehicle speed ($y_2$), on the other hand, might be from a call to a high level analytic. The fidelity does not demand knowledge of the amount of fuel consumed; rather only its sensitivity to speed.

For the emissions calculations in (3), these service calls would be calls to a large data model of engine performance. The complexity of the combustion process requires direct measurement from a test bed. These tests are at discrete system states (e.g. engine load and RPM) based on a design of experiments. Another service call would be needed to a utility for interpolating the data mesh.

Equation (2) exhibits the key features associated with making service calls. First, the (constant) acceleration $y_1$ may be called from a simple file or script for driver behavior. Next, the mass fraction of fuel converted to an emission typically would be derived from a large static data base of measurements from the engine test bed. The actual fraction of conversion must be interpolated from the measured data. Thus, two service calls are needed; one to the data base and one to the algorithm. Finally, the last service call for change of fuel consumption with respect to speed would likely be made from another simulation.

5 Conclusion and Future Challenges

In this paper we have illustrated how a ROSETTA framework can be utilized to provide analysis of vehicle emissions and performance as it performs a drive cycle. ROSETTA provides a rigorous, traceable framework to structure a workflow for a distributed simulation environment. ROSETTA is seen to provide a framework that extends the system structure model to dynamic simulation in a way that accounts for coupling and provides a verifiable analysis workflow that can be used for orchestration of services.

A major challenge with service-oriented simulation that we are current addressing is dealing with the changes in execution environments when providing a real-time integrated simulation capability. This will become more significant when hardware-in-the loop systems are integrated into the virtual simulation workflow, with a good example of this being a driver in the loop (DIL) simulation, requiring a real-time response.

6 References


