A WORKFLOW HYBRID AS A MULTI-MODEL, MULTI-PARADIGM SIMULATION FRAMEWORK[*]

Stuart Rossiter and Keith R.W. Bell
Department of Electronic & Electrical Engineering
University of Strathclyde
204 George Street, Glasgow, G1 1XW, UK
email: `stuart.p.rossiter@gmail.com`

KEYWORDS
hybrid simulation framework, social simulation, comparative, multi-paradigm, workflow

ABSTRACT

We propose that workflow software can be coupled with existing simulation frameworks (particularly agent-based ones) to provide three broad benefits: an improved modelling process due to the separation of concerns and rich scheduling syntax; interchangeable human and AI agents at minimal development cost; a common conceptual and software base for multi-model, comparative studies of the same system (including shared, distributed data visualisation).

We explain these benefits, providing a proof-of-concept framework implementation and examples from the domain of electricity generation expansion planning.

MOTIVATION

The complexity of social systems (e.g., markets) has led to a multitude of different paradigms and methodological schools for simulating them (Gilbert and Troitzsch 2005). Various software frameworks have sprung up to help structure this process, often with attempts to provide a more visual programming environment (e.g., the use of flowchart-driven behavioural processing in Repast Simphony (North et al. 2007), alongside traditional Java code). However, these typically assume a modelling paradigm (e.g., agent-based or system dynamics), and focus on structuring the model, supporting tools, performance and usability; the overall modelling *process* is left separate. (There are exceptions, such as Mimosa (Müller 2007), which is a suite of tools for a coherent modelling process formalism.)

This all means that studies of social systems are often multi-disciplinary, multi-model approaches where differences in concepts and modelling techniques mean it is very difficult for individuals to understand, compare and coherently extend existing models. (Axelrod (1997) provides a more focused summary on the scope and maturity of social simulation as a discipline.) As an example, consider the domain of **generation expansion models**. These attempt to model how investment decisions are made for electricity generation plant, and how they may fare in the resultant market; the aim being to consider what future configurations of plant might arise given certain exogeneous scenarios and endogeneous system mechanisms. Modern generation expansion models can vary from traditional least-cost optimisation (New Zealand Electricity Commission 2009), to aggregate systems-dynamics-based studies (Ford 1999), to detailed agent-based models with individualised decision-making (Botterud et al. 2007).

This domain is interesting because, despite these differences, the particular nature of electricity as a product—and its associated technical infrastructure—means that participant behaviour actually has some well-defined constraints, in terms of the electricity market protocols and generation connection processes enforced by the transmission network operator. (These constraints can be contrasted with the idiosyncratic, per-organisation processes by which generating companies make their strategic and operational decisions.) We call these **constraining global processes** hereafter.

A WORKFLOW HYBRID FRAMEWORK

Workflow software is normally used to help define, automate and re-engineer business processes. The workflows themselves are defined using some specification language, which typically maps to some graph-based formalism. They run in an engine and, where human input is required for tasks, users interact with the workflow via some front-end (see figure 1). Automated 'users' may also be responsible for some workflow tasks, or for custom code which might do things like allocating a user to a task, or deriving and rendering specific data.

We propose that workflow software can be used in a framework for simulation models which primarily use another paradigm (e.g., agent-based); effectively, the workflows de-

---

[*] This is a post-print version of the article published with the following citation:

Rossiter S. and Bell K., 2010. *A workflow hybrid as a multi-model, multi-paradigm simulation framework.* In G. Janssens; K. Ramaekers; and A. Caris (Eds.), *ESM 2010: The 2010 European Simulation and Modelling Conference*. Eurosis, 37–41.

The definitive published version is available from the publisher (Eurosis): see the conference page at `http://www.eurosis.org/cms/index.php?q=taxonomy/term/21`. The main author's email address has been updated.

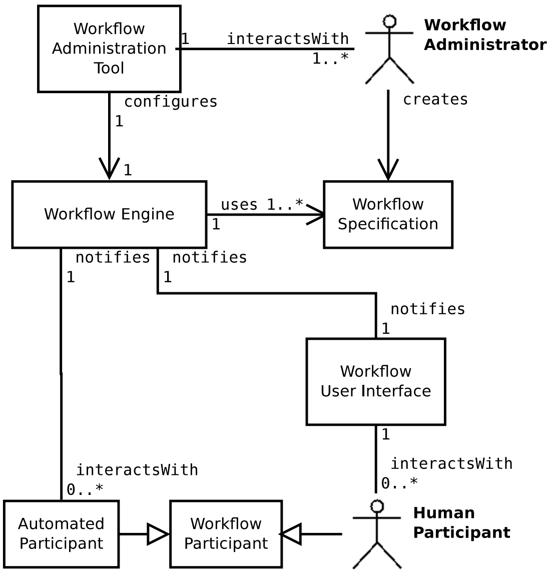

Figure 1: Generic workflow software architecture (as a UML class diagram, with actor icons used for classes representing human participants)

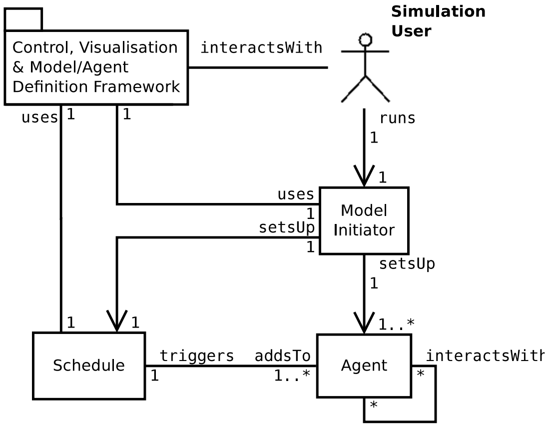

Figure 2: Generic agent-based modelling (ABM) architecture (same notation as figure 1)

fine selected scheduling and related data definitions. We argue that such a hybrid can help address some of the issues in the previous section, despite the fact that, on the surface, it seems very much a 'like-for-like' replacement of the scheduling mechanism already used. The benefits are particularly apparent for multi-model studies of systems which have constraining global processes, since decisions on what processes to enshrine in workflow form are much more apparent and 'uncontroversial'.

The hybrid framework includes a **workflow framework**, and a coupled **main model framework**. Such hybridisation fits best conceptually for *agent-based* main model frameworks, since there is a natural fit between workflow users and agents in the agent-based model (ABM), and typically a clear separation of scheduling from agent behavioural logic (cf. system dynamics as a set of coupled difference equations). Workflows also tie their actions to roles, which aligns nicely with the use of object-oriented inheritance in agent development. However, we believe it is still feasible for other paradigms (see later). For reference, the basic ABM architecture is shown in figure 2, with the assumption that no further background explanation is needed.

We define three areas of potential benefit in the sections below.

Improved Modelling Process

Using workflows provides a natural, visual development style which enforces early, up-front consideration of the overall control flow (in terms of real-world processes), attendant data structures, and the set of agent roles involved. (Most workflow software uses a platform-neutral XML data representation, which also helps move the modeller away from programming language specific design.) This therefore enshrines some good model design practices, including the separation of this 'programming-in-the-large' from the 'programming-in-the-small' of the agent behaviour; a software improvement technique with a long history (DeRemer and Kron 1975). Although similar separations and visual environments may exist in ABM frameworks, their use is much more optional than in this approach. In addition, workflows provide a rich set of control primitives which may be better suited for clearer and more powerful scheduling.

In terms of model extension, reuse and collaboration, this separation also helps allow different agent implementations to be switched in and out, since there is no temptation to embed control flow logic in the agents themselves, and the data interfaces for each agent action are clearly and centrally defined. We have also found in practice that having to focus on shared data definitions for multiple models is very effective in clarifying the essential concepts and meanings that the data embeds.

Finally, agents often tend to require information on the actions of other agents to update their own internal models. The decoupling above can be enhanced by defining what we call **informational events** in the workflow. That is, 'real' workflow tasks (carried out by agents) can be followed by 'infrastructural' tasks which publish events providing data on the action that just occurred and its outcomes. Agents can subscribe to events of interest, thus decoupling such informational interactions from agent logic and further centralising agent interaction capabilities. (A modeller can then choose to implement their own information filtering logic to reflect things like interaction topologies, without having this fixed by interaction logic encoded elsewhere.) See the *Info Events Server* component in figure 3.

Interchanging Human and AI Agents

It is very difficult to validate models of human decision-making, primarily because the modeller cannot easily experiment with the real-world system (Windrum et al. 2007). This means that empirical simulation experiments can be very useful, where real humans may take the place of, or compete against, computational (AI) agents (Tesfatsion 2002; p.57). Human agents can also be used for computational steering, directing the simulation towards interesting areas of behaviour. If a modeller wants to be able to mix or interchange human and AI agents, most of the complication is in the data definition, rendering, thread control and user interface for the human agent. However, workflow software is designed precisely to handle these aspects, so we can get these benefits at minimal development cost, without some custom-coded client-server simulation—such as PSERC's PowerWeb simulation of power exchange auction markets (Zimmerman et al. 1999).

There are some inherent complexities in merging workflow-based human actions (which normally run in real-time) with a simulation-time-based ABM framework (see later).

A Common Conceptual and Software Base for Multi-Model Research

If we want to use *shared* workflows to add some structure and coherency to multi-model approaches, the question becomes what should be 'workflow-ised' without overly restricting the modelling freedom required. Do shared workflow definitions even make sense where we may be modelling the same real-world system at different aggregation levels, and potentially focusing on different aspects?

We propose that it *does* make sense for for real-world systems which have constraining global processes, as discussed earlier. The use of workflows brings the following conceptual and development benefits:

1. It explicitly models baselines for the identified constraining processes, where the level at which the processes are defined establishes up-front what type of freedom the set of models is intended to have. It also specifies the aggregation level at which some form of comparison is likely to be required.

   Models which use these workflows are ensured as consistent at this conceptual level. Each may model further, disaggregated detail *outside* of the workflow definitions. (For example, the workflow might represent a daily market as a 'black box', with details of participants passed in, and per-participant revenues output; particular models would use their own sub-model to determine what those output revenues were.)

   In practice, we also found that the need to tie-in with the workflows makes it much clearer to the modeller when their particular model is moving away from the conceptual underpinnings, since they find themself having to code around 'restrictions' in the design. This forces a deeper consideration of where the conceptual discontinuity lies, which is very useful in effectively comparing and discussing models.

2. It provides a consistent terminology and set of causal assumptions on the system (including assumptions such as what order elements of these processes have to occur in).

3. By introducing the possibility for human agents, it directs the modeller to produce meta-data with maximal cross-model, long-term benefit. To see why this is true, consider that human agents typically prefer human-oriented graphical or statistical aggregate representations of data. In addition, most decisions will often refer back to a small number of 'global' data items which tend to reflect the shared environment within which the agents operate (such as, for generation expansion models, underlying plant costs or power flows on the electricity transmission network under certain load or outage scenarios).

   Therefore, part of our framework includes a standard way to define these global fields, their statistics or visualisations, and human/AI agent access to them. (Agents can still filter this data to model aspects such as perceptual range outside of the workflows.) Workflow software typically provides a generic user-interface (e.g., a visual representation of an XML data hierarchy), with software 'hooks' to allow for customised visualisation as required. Such customisations can reuse the visualisation capabilities of the coupled main model framework, though this requires some distribution of these capabilities so that visualisations can be recreated for human agents participating in the model via networked clients (see figure 3).

   Importantly, these data visualisations also tend to be useful views on the dynamics of the model *for the simulation researcher*, so the modeller is not significantly wasting coding effort if human agents do not end up being extensively used. Because this meta-data is based on well-thought-out underlying data (that has been agreed as a consistent base for a set of possibly very different models), major future changes should not be needed, and the code provides clean separation of shareable meta-data from agent decision logic.

4. It ensures careful consideration of the effects of differing aggregation levels, which is often the main distinction between modelling paradigms (e.g., Bonabeau (2002) compares agent-based and system dynamics representations of the same system).

   Workflows are typically defined at the lowest aggregation level (e.g., individual traders in a market), with the coupled main model having to aggregate and disaggregate as it requires. This makes it much easier to do comparative models that compare the effects of individual variation at differing definitions of ‘individual’.

PROOF-OF-CONCEPT IMPLEMENTATION

HAWSER (Hybrid Agent-Workflow Simulation Engine for Research) has been developed as a proof-of-concept implementation of the framework. (Some specific aspects, such as distributed visualisation, are not yet implemented.) HAWSER couples two existing, open source frameworks: MASON (Luke et al. 2005) for agent-based modelling, and YAWL (van der Aalst et al. 2004) for workflows. All code, including both open source frameworks, is written in Java.

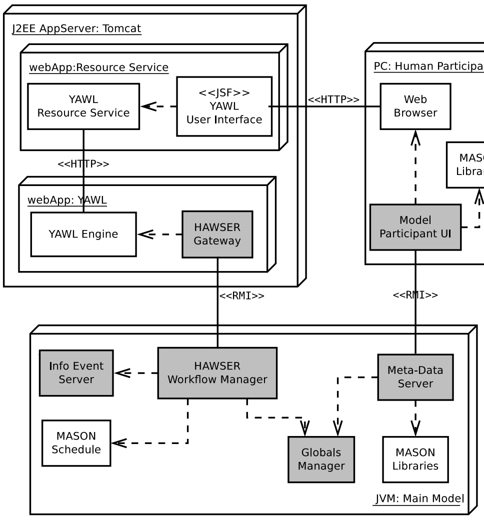

Figure 3: High-level deployed architecture of the HAWSER framework (as a UML deployment diagram). Shaded components are those added to the existing ABM and workflow frameworks

The high-level architecture is shown in figure 3, which previous discussions refer to. Due to space restrictions, we do not describe this further, restricting ourselves to an overview of the key technical challenges. We intend to publish the framework, together with detailed documentation and demo models, as an open source project in the near future.

Technical Challenges

1. The main ABM runs in a separate process (Java virtual machine) to the YAWL system, and thus inter-process communication is required. We use Java RMI for simplicity. This also allows the workflow software to run on a separate physical machine if required.

   YAWL provides the ability to define a third-party (‘observer’) gateway which receives notification of all new workflow tasks and has direct access to the main YAWL engine. We use this feature to implement a HAWSER Gateway component, which communicates with a Workflow Manager component within the main model’s process. The latter provides the bridge for workflow actions to be passed on to the relevant agents, and is used to initiate workflows.

2. The MASON discrete-event schedule defines the global simulation clock. Workflow tasks, which now represent agent actions, should therefore take some elapsed simulated time to complete; that is, the action should not be completed until a simulation timestep which represents the simulated completion time.

   Agents declare the simulated time taken, and special Deferred Action Handler objects are added to the schedule to trigger task completion at the appropriate timestep.

3. Workflows with parallel tasks result in multiple, parallel threads which need to be controlled so that they are fed *sequentially*, and in a *repeatable order* to agents. (It is a fundamental simulation requirement that models are repeatable, in that the same model with the same input data, including a random seed for stochastic elements, results in exactly the same output.) This is handled via meta-knowledge of what tasks are going to be, or have been, triggered by workflow actions. All new threads are held by the Gateway, and the meta-data used to release tasks in a deterministic order.

   In addition, the main ABM model has to run with only a single thread active at any one time. This requires some elaborate control logic in the Gateway and Workflow Manager to determine when main model threads should wait or continue (especially since workflows may be run from within workflows).

4. To avoid excessive data traffic to and from workflows, it often makes sense to visualise global data externally to the workflow software, but presented in a common unified interface to human agent participants. The framework is designed to support this, which also allows AI agents to be presented directly with the more powerful object-based representations (cf. some ‘flattened’ XML equivalent in workflow data).

5. Agents may wish to delay their processing (within some time window) to see what others agents do. (What they are actually aware of is dependent on the specifics of the model.) This complicates matters for human agents, who need to be able to declare that they are waiting for some simulated time. Enabling such time limits and human re-processing loops requires dynamic alteration of special template workflow definitions. YAWL does not currently support this, but work to add such a capability is planned.

Generation expansion models have been realised using the framework and the related multi-model methodology. The global data consists of a representation of the transmission network, together with visualisations useful to the researcher and human agents.

REFLECTIONS & FUTURE WORK

We reflect on two general aspects of this work.

Theory & Novelty

We are promoting the benefits of a workflow formalism for agent scheduling, and we believe that there are significant methodological benefits for the right types of real-world system. In the workflow literature, and the related field of queueing models in operational research, workflow *has* been coupled with simulation (and agents), but the interest has been in the workflows *themselves*, and the business processes they represent—such as in: the simulation *of* workflows to test proposed business process changes (Rozinat et al. 2008); ABM alternatives to queueing models to represent dynamic business processes (Tan et al. 2007); and workflows representing inter-agent interaction frameworks (Zhuge 2003). This is rooted in the history of workflow software as a tool for business process optimisation and re-engineering; we are 'repurposing' workflows for 'normal' simulation modelling. From a theoretical point of view, we should bear in mind that *all* simulations with discretised time can be represented by some form of discrete-event based model (Zeigler et al. 2000), and that this DEVS formalism can be extended to explicitly represent agent-based models (Müller 2009). The HAWSER implementation aligns with the common ABM practice of using randomised iteration to represent simultaneous events. Müller (2009) points out that this is a potentially undesirable formalism, and looks at various DEVS extensions to provide a better alternative. It may therefore be of interest to further consider workflow-ABM coupling in this more theoretical light, particularly as workflow formalisms are specifically designed to model concurrent processes.

Extension to Non-ABM Models

We stated earlier that, in theory, the approach could be used for other simulation paradigms, notably system dynamics (the DEVS background discussed above supports this). However, the details need to be considered and proven: workflow data has to link to and from the stocks and flows of the system dynamics paradigm. Ninios et al. (1995) discuss some of the 'paradigm clash' difficulties in making such a switch.

REFERENCES[1]

Axelrod R., 1997. *Advancing the Art of Simulation in the Social Sciences*. In R. Conte; R. Hegselmann; and P. Terna (Eds.), *Simulating Social Phenomena*, Springer, *Lecture Notes in Economics and Mathematical Systems*, vol. 456. 21–40. URL `http://www-personal.umich.edu/~axe/research/AdvancingArtofSim.pdf`.

Bonabeau E., 2002. *Agent-based modeling: Methods and techniques for simulating human systems*. *Proceedings of the National Academy of Sciences*, 99, 7280–7287. URL `http://www.pnas.org/content/99/suppl.3/7280.short`.

Botterud A.; Mahalik M.; Veselka T.; Ryu H.; and Sohn K., 2007. *Multi-Agent Simulation of Generation Expansion in Electricity Markets*. In *IEEE Power Engineering Society 2007 General Meeting*. IEEE, 1–8.

DeRemer F. and Kron H., 1975. *Programming-in-the large versus programming-in-the-small*. In *Proceedings of the international conference on Reliable software*. 114–121.

Ford A., 1999. *Cycles in competitive electricity markets: a simulation study of the western United States*. *Energy Policy*, 27, 637–658.

Gilbert N. and Troitzsch K., 2005. *Simulation for the Social Scientist*. Open University Press, 2nd ed.

Luke S.; Cioffi-Revilla C.; Panait L.; Sullivan K.; and Balan G., 2005. *MASON: A Multiagent Simulation Environment*. *Simulation*, 81, no. 7, 517–527.

Müller J., 2007. *Mimosa: using ontologies for modeling and simulation*. In *Proceedings of the 8th Asia-Pacific complex systems conference (Complex'07)*.

Müller J., 2009. *Towards a Formal Semantics of Event-Based Multi-agent Simulations*. In *Multi-agent Based Simulation IX*. Springer, no. 5269 in LNCS, 110–126.

New Zealand Electricity Commission, 2009. *2009 Grid Planning Assumptions*. Tech. rep., New Zealand Electricity Commission. URL `http://www.electricitycommission.govt.nz/consultation/09-gpa`.

Ninios P.; Vlahos K.; and Bunn D., 1995. *Industry Simulation: System modelling with an object oriented / DEVS technology*. *European Journal of Operational Research*, 81, 521–534.

North M.; Tatara E.; Collier N.; and Ozik J., 2007. *Visual agent-based model development with Repast Simphony*. In *Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence*.

Rossiter S. and Bell K., 2010. *A workflow hybrid as a multi-model, multi-paradigm simulation framework*. In G. Janssens; K. Ramaekers; and A. Caris (Eds.), *ESM 2010: The 2010 European Simulation and Modelling Conference*. Eurosis, 37–41.

Rozinat A.; Wynn M.; van der Aalst W.; ter Hofstede A.; and Fidge C., 2008. *Workflow Simulation for Operational Decision Support Using Design, Historic and State Information*. In *Proceedings of the 6th International Conference on Business Process Management (BPM 2008)*. Springer-Verlag, 196–211.

---

[1] Due to a quirk of the LaTeX packages used to prepare this post-print version, these references include an entry for the paper itself! Needless to say, this is not in the published version.

Tan W.; Li S.; Tang A.; and Shen W., 2007. *A Workflow Simulation Framework Based on Multi-agent Cooperation*. In *Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design*.

Tesfatsion L., 2002. *Agent-based computational economics: growing economies from the bottom up*. *Artificial Life*, 8, 55–82.

van der Aalst W.; Aldred L.; Dumas M.; and ter Hofstede A., 2004. *Design and implementation of the YAWL system*. In *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAISE 2004)*. 281–305. URL `http://www.springerlink.com/index/cpa194xbmauduuwn.pdf`.

Windrum P.; Fagiolo G.; and Moneta A., 2007. *Empirical Validation of Agent-Based Models: Alternatives and Prospects*. *Journal of Artificial Societies & Social Simulation*, 10, no. 2, 8. URL `http://jasss.soc.surrey.ac.uk/10/2/8.html`.

Zeigler B.; Gon Kim T.; and Praehofer H., 2000. *Theory of modeling and simulation : integrating discrete event and continuous complex dynamic systems*. Academic Press, 2nd ed.

Zhuge H., 2003. *Workflow- and agent-based cognitive flow management for distributed team cooperation*. *Information & Management*, 40, no. 5, 419–429.

Zimmerman R.; Thomas R.; Gan D.; and Murillo-Sánchez C., 1999. *A Web-based platform for experimental investigation of electric power auctions*. *Decision Support Systems*, 24, 193–205.