Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 88 (2016) 537 – 543

Energy Procedia

CUE2015-Applied Energy Symposium and Summit 2015: Low carbon cities and urban energy systems

Robustness diagram with loop and time controls for system modelling and scenario extraction with energy system applications

Jasmine Malinao[a,c,*], Florian Judex[a], Tim Selke[a], Gerhard Zucker[a], Henry Adorna[b], Jaime Caro[b], Walter Kropatsch[c]

[a]*Energy Department, AIT Austrian Institute of Technology, Vienna, Austria*
[b]*University of the Philippines, Diliman, Quezon City, Philippines*
[c]*Vienna University of Technology, Vienna, Austria*

Abstract

In this research, we introduce an extension of Robustness Diagrams for modelling complex systems such as energy systems. We provide a construction scheme of this extension for the inclusion of looped and time-dependent substructures for an effective modelling framework. The latter set of substructures is introduced in this work as "reset-bound subsystems". We introduce a scenario extraction algorithm to obtain behavioral profiles from the models. Lastly, we apply our scheme to create a model of a real-world energy system and use the proposed algorithm to extract a scenario describing one process done by the system being modelled.

Peer-review under responsibility of the organizing committee of CUE 2015

Keywords: Energy systems; Graph reductions; Robustness Diagram; Workflows

1. Introduction

Workflow models, as formulated by van der Aalst[1], are case-driven representations of processes that are formulated under three dimensions namely, i) process, ii) resource, and iii) case. Some of the models where the process dimension is used are Petri nets by Murata[2], workflow nets by van der Aalst[1, 3], Spiking Neural networks by Cabarle et al[4], among others. Meanwhile, Class and Use Case Diagrams of the Unified Modelling Language(UML) from Rosenberg[5] are examples of models using the resource

* Corresponding author. Tel.: +43 (0) 50550-2032; fax: +43 (0) 50550-2037
E-mail address: jasmine.malinao@ait.ac.at

Peer-review under responsibility of the organizing committee of CUE 2015
doi:10.1016/j.egypro.2016.06.075

and case dimensions, respectively. However, these models (and including others in literature) are only specifically constructed using one or two dimensions. Therefore, users are not provided a holistic view of the various components and processes composing the activities that a system undertakes in its operation. It would become more difficult and cumbersome for users to perform repeated cross-referencing of multiple, multi-type sub-models as the complexity of the system grows. Nevertheless, the Robustness Diagram(RD) of the UML as proposed by Rosenberg et al.[6] provides partial support for modelling using all three dimensions. Moreover, the algorithm based in RDs which was introduced by Malinao et al. [7] which extracts a scenario, i.e. a description of system functionality with temporal information, from RDs has the following weaknesses: a) there is no distinction between sequential tasks which occur with prerequisites and those which do not have. The execution of the second task in the sequence is needlessly suspended where the suspension may trigger faults and failure in a complex system; and b) the absence of the mechanism to redo a task or a group of task wherein the repetition may be time-dependent in real-world systems.

In this work, we propose the **RD with loops and time controls**(RDLT) to address the aforementioned problems. RDLTs provide users to create a model with all three workflow dimensions in use. We provide a construction scheme for RDLTs to enable this integration of the dimensions in a workflow. We also propose "reset-bound subsystems" in RDLTs to include volatile components in modelling. With these, we revise the scenario extraction algorithm in [7] to effectively extract scenarios since RDLTs are now multi-input, multi-output, and multi-case representations of real-world complex systems. Finally, we use RDLTs to represent an instance of a complex system, i.e. an energy system[8], and extract a scenario from this model.

2. RD with loop and time controls

DEFINITION 1. A *Robustness diagram with loop and time controls(RDLT) is an 8-tuple* $R' = (V1, V2, V3, E, C, T, L, M)$ *where:*

1. $V1$ *are boundary objects,* $V2$ *are entity objects,* $V3$ *are controllers, where* $V1 \cap V2 \cap V3 = \text{null}$,
2. $E \subset ((V1 \cup V2)\times V3) \cup (V3 \times (V1 \cup V2 \cup V3))$ *is a set of arcs,*
3. $C : E \rightarrow \{\Sigma \cup \{\lambda\}\}$ *is a multiset of labels, where* Σ *is a finite non-empty set of symbols, and* λ *is the empty string. For any* $(x,x) \in E$, $x \in V3$, $C((x,x)) = \lambda$.
4. L *assigns the maximum number of traversals allowed for each arc in* R', *where* $L(.) \geq 1$.
5. $T : E \rightarrow [ti]^n$ *assigns a nonnegative value ti to* (x,y) *marking the time at the ith traversal on* (x,y) *using an algorithm's walk in* R', $i = 1, 2, \ldots, n$, $n = L((x,y))$. *All values of T are initially set to* $[0]^n$.
6. $M : V1 \cup V2 \rightarrow \{0, 1\}$ *is a (user-defined) value of an object* $u \in V1 \cup V2$ *to indicate whether a subgraph of* R' **with center** u, *denoted as* $Gu \subseteq R'$, *resets its values of T under some reset process (to be discussed later). The vertex set VGu(which includes u) of Gu's, and edge set EGu includes* $v \in V3$ *if* $(u,v) \in E$ *and* $C((u,v)) = \lambda$, *and* $(a,b) \in EGu$ *if* $a,b \in VGu$, *and* $(a,b) \in E$. *If* $M(u) = 1$, *we call Gu a* **reset-bound subsystem** *of* R'.

2.1. On edge traversals and time updates

We revise the scenario extraction algorithm *Alg* from Malinao et al.[7] for the RDLT R'. We propose Alg' with new sets of conditions for traversals and updates of the new metadata found in R' as follows,

(1) For a node $x \in V1 \cup V2 \cup V3$, Alg' performs a **check** on x. Checking considers which outdegrees of x can be selected for traversal based on the former's values of T and L. An outdegree $(x,o) \in E$ is included in the selection $CA(x)$ if we have $T((x,o))=[0]$ or $\exists i \leq L((x,o))$, $ti\text{-}1 > ti$, for $T((x,o)) = [t1,\ldots,tL((x,o))]$, $o \in V1 \cup V2 \cup V3$. From $CA(x)$, one of them is arbitrarily selected for a possible traversal. Let $(x,y) \in CA(x)$ be this selected outdegree. Finally, the check process updates $T((x,y))$(refer to Section 2.1.(2)). Then, Alg' tests if traversal on (x,y) is possible. $(v,y) \in E$, $v \in V1 \cup V2 \cup V3$, does not prevent the traversal on (x,y) if any of the following holds, (a) $C((v,y)) \in \{\lambda, C((x,y))\}$, (b) $|\{ti \in T((x,y)) | ti \geq 1\}| \leq L((v,y)) \wedge T((v,y)) \neq [0]$, (c) $C((v,y)) \in \Sigma, C((x,y)) = \lambda \wedge T(v,y) \neq [0]$.

We call (x,y) **unconstrained** if $\forall (v,y) \in E$, traversal on (x,y) is possible. If (x,y) is unconstrained, Alg' traverses it. The process of checking and traversal is repeated on y (and all nodes possible) until the arc f is checked. On another hand, if (x,y) is not unconstrained, f has not been traversed yet, and $|CA(x)| = 0$, Alg' backtracks from x. Backtracking means Alg' goes back to the nearest ancestor u of x, $|CA(u)| > 0$, and proceeds from there.

(2) values for $T(.)$ of R' shall be (re)set as follows,

(a) from the starting point $s \in V1 \cup V2$ of the walk done by Alg', arbitrarily choose $d \in V3$, where $(s,d) \in E$ to consider adding to the walk. Set $x = s$ and $y = d$.

(b) When Alg' checks (x,y), set $ti \in T((x,y))$, for $ti\text{-}1 > ti$, $i \in \{1, 2,\ldots,L((x,y))\}$, as follows,
$ti = \max_{\forall u,(u,x)\in E}\{\max_{k=1,2,\ldots,L((u,x)), tk\in T((u,x))}\{tk\}\}+ 1$, if $\exists (u,x) \in E$, 1 otherwise.

Alg' traverses (x,y) if it is unconstrained. With the update of $T((x,y))$ and the traversal of (x,y), we evaluate further the remaining indegrees of y to see if their T are consequently updated or not. Let $(z,y) \in E$. We either retain or update all other values of $T((z,y))$ based on the following rules, as

- if $C((x,y)) \neq \lambda$, update the values of T of all remaining indegrees of y except those labeled with λ. Reset $tj = ti$, $tj \in T((z,y))$, where $tj = \max_{k=1}^{L((z,y))}\{t k\}$, $\forall z, (z,y) \in E$, where $C((z,y)) \neq \lambda$.
- if $C((x,y)) = \lambda$, retain $T((z,y))$, $\forall z, (z,y) \in E$.

(c) If (x,y) is traversed, Alg' evaluates if for some $u \in V1 \cup V2$, Gu is a reset-bound subsystem of R' where $x \in VGu$ and y is not in VGu (and hence (x,y) is not in EGu). If Gu is a reset-bound subsystem, Alg' sets $T((a,b)) = [0]$, $\forall (a,b) \in EGu$. Moreover, suppose (x,y) is traversed where (x,y) connects a reset-bound system Gu to another reset-bound subsystem Gz of R', where $x \in VGu$, $y \in VGz$, we recommend that the user sets $y = z$ in R'. Assigning $y = z$ assures that any input $a \in \Sigma$ which is required by Gz from Gu is effectively passed on through x, i.e. $C((x,y)) = a$.

If Alg' has reached and updated T of the arc f, it terminates.

2.2. On deadlocks and reachability in RDLTs

DEFINITION 2. *Let* $R = (V1,V2,V3,E,C,T,L,M)$ *be a RDLT. Let* $xi \in V1 \cup V2 \cup V3$, $i \in \{0, 1,\ldots,n - 1\}$, $n=|V1 \cup V2 \cup V3|$. *Assign the weight* $W(xj,xi) = 1/k$ *to* all (xj,xi), *i.e.* $(xj,xi) \in E$, $xj \in V1 \cup V2 \cup V3$, *where k is the total number of indegrees of xi.*

Let $x0 \in V1 \cup V2 \cup V3$ *be a user-defined start node of R, where* $\forall xk \in V1 \cup V2 \cup V3$, $\nexists (xk,x0) \in E$.

Let $TS(x0 : xi,A)t$ *be the sum of weights* $W(.)$ *of the edges traversed by an algorithm A at time* $t \in lN$, *from its initial step on* $x0$ *to its goal xi in R. Initially,* $TS(x0 : xi,A)0 = 0$.

We say that xi is **reachable** *from* $x0$ *if* $\exists V p \subseteq V1 \cup V2 \cup V3$, $x0, xi \in Vp$, *if* $TS(x0{:}xi,A)t = TS(x0 : xi,A)t\text{-}1 + \sum\forall(xr,xs) \in E,xr,xs \in VpA(W(xr,xs))$, *where* $1 \leq t \leq diam(Vp)$, $diam(Vp)$ *is the diameter of Vp, and* $A(W(xr,xs)) = W(xr,xs)$ *when A traverses* (xr,xs), $\forall xr \in Vp$, *at time t, 0 otherwise. When xs is reached by A, we consider that* $\forall xr \in V1 \cup V2 \cup V3$, (xr,xs) *are traversed simultaneously at time t, hence,* $\sum\forall(xr,xs) \in E,xr \in VpA(W(xr,xs)) = 1$, *or* $TS(x0{:}xi,A)diam(Vp) = |Vp|\text{-} 1$.

Let $RVp = (V1',V2',V3',E',C',T',L',M')$ *be the vertex-induced subgraph of R using Vp, i.e.* $V1' \subseteq V1$, $V2' \subseteq V2$, $V3' \subseteq V3$, *hence* $Vp = V1' \cup V2' \cup V3'$, $E' \subseteq E$, *and the sets* C',T',L',M' of *Rvp are mappings consistent with E,C,T,L,M of R. RVp is* **deadlock-free** *wrt A if* $TS(x0 : xi,A)diam(Vp) = |Vp|\text{-} 1$.

THEOREM 1. *Given any RDLT* $R = (V1,V2,V3,E,C,T,L,M)$, *if* $\exists xa, xd \in V1 \cup V2 \cup V3$, *where xa is an ancestor of xd and* $(xd,xa) \in E$, $C((xd,xa)) \in \{\lambda\} \cup \{\beta\}$, *where* $\beta = C((xs,xa))$ *if* $C((xr,xa)) = C((xs,xa)),\forall(xr,xa) \in E$, λ *otherwise, and there is no path from* $x0$ *to xd not using the arc/s* $(xs,xa) \in E$, $\forall xs \in V1 \cup V2 \cup V3$, *then R is not deadlock-free and any descendant of xa (including xa) are unreachable using Alg, however, R is deadlock-free using Alg'.*

3. On applying the proposed framework on energy systems

Adsorption chillers are designed as closed-cycle (vacuum) machines with an evaporator, adsorber/desorber reactor beds, and a condenser wherein the refrigerant is passed throughout these components during cycles of operation of chillers. Associated to these components are three main temperature loops circuits, i.e. heating water, chilling water, and cooling water. These loops aid in circulating the refrigerant throughout all the components of the chiller by the processes of evaporation, adsorption/desorption, and condensation. The chiller's midsection provides two reactor beds which continuously alternate as an adsorber/desorber throughout the machine operation for a stable and continuous cooling. Adsorption takes place as the refrigerant heated and vaporized in the evaporator. By pressure value differences of the evaporator and the reactor bed, a valve is opened thereby transporting the vaporized refrigerant to the adsorber bed. Simultaneously, the other reactor bed desorbs the previously-adsorbed refrigerant. In a similar manner, opening of another valve transports the desorbed refrigerant to the condenser. In the condenser, the refrigerant is liquified and is sent back towards the evaporator to close the cycle. Before the two beds switch roles as adsorber and desorber, the system of pneumatically-actuated valves open and close to connect or disconnect appropriate chambers of the chiller to enable refrigerant mass and heat recovery periods. (For more details of the chiller, see the descriptions provided by Rezk[8].)

3.1. RDLT of the adsorption chiller

Figure 1 shows an RDLT R′ modelling the cited adsorption chiller and its processes. It shows markings of T representing one of the many scenarios which we can extract from R′. In this model, the extracted scenario starts when the chiller is activated by starting at the boundary object 'a1' (at time step 1) and ends when the liquid refrigerant exits the condenser and flowing back to the evaporator. Essentially, the extracted scenario models the refrigerant flow from its stages of evaporation, desorption/adsorption, and condensation. The final arc which was traversed by the algorithm that represents this terminal process of this scenario is ('h7', 'e1') at time step 31. All other intermediate nodes which the algorithm walked through are annotated in red font their corresponding time of traversals.

The resets of *T* (or the absence thereof) on arcs which are not included in reset-bound systems but when traversed triggers the reset, e.g. ('f3', 'e7') at time step 12, is an important aspect in effective scenario extraction and system modelling. Notice that the algorithm does not perform traversal ('f3', 'e7') at any moment during the time steps {20, 21} and {29, 30} since we obtain *T*(('f3', 'e7')) = *L*(('f3', 'e7')) = 1 already. This correctly models the real-world and thermodynamically-bound operation of the chiller. Had we set *L*(('f3', 'e7')) > 1, a wrong control scheme would have proceeded after time step 12. This is one of the main reasons why a scheme for loops and resets are disjointly formulated for modelling various systems.

With the application of our proposed algorithm on *R′* in Figure 1, we obtain the scenario: 'a1 a2 [c1, b1, d1] b2 e1 [e2 e3] e4 e5 e6 (f1 f2 f3) e7 g1 [g2, g3, g5] g4 g6 g7 (f1 f2 f3) g8 h1 [h2, h3] h4 h5 h6 (f1 f2 f3) h7'. This sequence shows the order of reachability and execution of the nodes of the RDLT, i.e. from 'a1', the algorithm reaches and executes 'a2' at time step 1. The nodes enclosed in square brackets [ ] represent nodes which are reachable at the same time step by parallel flows of execution from its parent node (e.g. parallel flows from from 'a2' to either 'c1' or 'b1' or 'd1'). The nodes enclosed in parentheses represent those which belong to reset-bound systems.

4. Conclusions

In this research, we proposed an extension of RDs called RDLTs to accommodate multi-input, multi-output, and multi-case modelling of complex systems. In particular, RDLTs supports scenario representation and extraction which may require localized component reuse and time-dependent subsystem usage.we proposed an extension of RDs called RDLTs for complex systems modelling. In particular, RDLTs supports scenario representation and extraction which may require localized component reuse and time-dependent subsystem usage. Furthermore, we provided an algorithm to effectively extract scenarios in RDLTs. Finally, we showed an application of RDLTs on energy systems.

References

[1] van der Aalst, WMP. Structural Characterizations of Sound Workflow Nets. *Computing Science Reports 96/23.* Eindhoven University of Technology; 1996.

[2] Murata, T. Petri Nets: Properties, analysis and application. In: Proceedings of the IEEE; **77**(4):541-580; 1989.

[3] van der Aalst, WMP. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems and Computers*. Vol. **8**(1):21-66; 1998.

[4] Cabarle, FG, Adorna H. On Structures and Behaviors of Spiking Neural P Systems and Petri Nets. In: Proc. in the International Conference on Membrane Computing; 2012, p. 145-160

[5] Rosenberg, D, Stephens, M. Use Case Driven Object Modeling with UML: Theory and Practice. Berkley: Apress. ISBN-10: 1590597745; 2007.

[6] Rosenberg, D., Stephens, M. & Collins-Cope, M.: Agile Development with ICONIX Process; 2005.

[7] Malinao J, et. al. A Metric for User Requirements Traceability in Sequence, Class Diagrams, and Lines-of-Code via Robustness Diagrams. In: Proc. on Information and Communications Technology; Springer Japan, ISBN 978-4-431-54435-7, 2013, p. 50-63

[8] Rezk A. Theoretical and Experimental Investigation of Silica Gel/Water Adsorption Refrigeration Systems", PhD Dissertation, University of Birmingham; 2012

Biography

Jasmine Malinao is engaged in basic and applied research on complex systems modelling, graph representations, data mining and data signatures, algorithmics, design and implementation, and requirements traceability for software engineering. She obtained her bachelor and masters degree in Computer Science from the University of the Philippines in the Philippines. She is currently doing her PhD in Computer Science in Vienna University of Technology in Austria. She works as a freelance researcher in AIT Austrian Institute of Technology in Vienna, Austria.

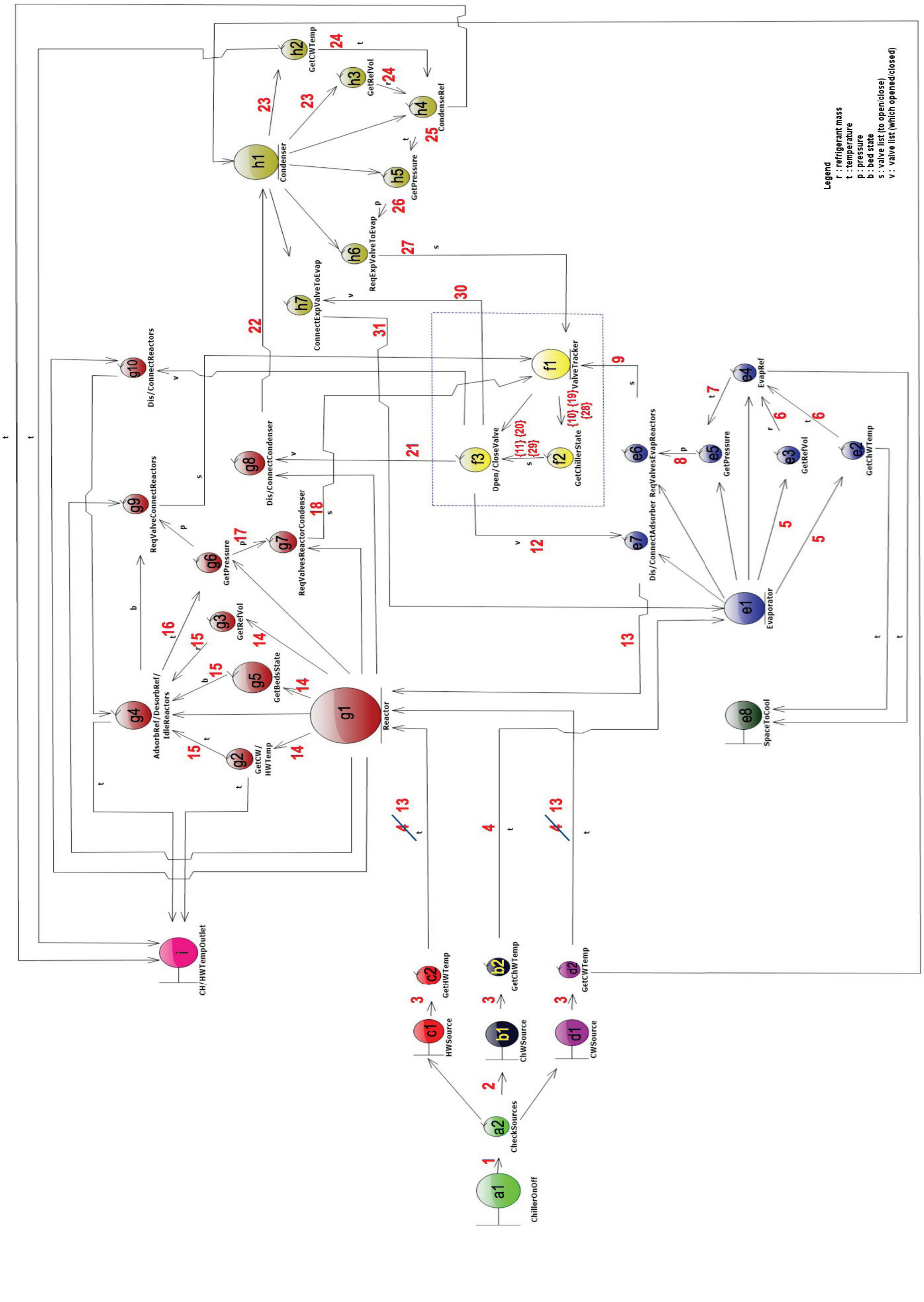

Fig. 1. Robustness Diagram with Loops and Reset Controls Representation of a 2-bed adsorption system. Here, $M(v) = 0$, $\forall v \in \{V_1 \cup V_2\} \backslash \{'f1'\}$, and $M('f1') = 1$.