

GREEN BUSINESS PROCESS MANAGEMENT: A RESEARCH AGENDA

Aditya Ghose, Konstantin Hoesch-Klohe, Lothar Hinsche and Lam-Son Le

Decision Systems Lab
School of Computer Science and Software Engineering, University of Wollongong
Wollongong 2522, Australia

ABSTRACT

There is a global consensus on the need to reduce our collective carbon footprint. While much research attention has focused on developing alternative energy sources, automotive technologies or waste disposal techniques, we often ignore the fact that the ability to optimize (existing) operations to reduce their emissions impact is fundamental to this exercise. Business process management (BPM) technology, with its focus on understanding, modelling and improving/optimizing business processes, is a key starting point. Process modelling technology has applications beyond what we would traditionally describe as business processes - we can also model and improve manufacturing and other "physical" processes. This paper describes the contours of the emerging research landscape in green business process management and presents some early results in this area.

INTRODUCTION

There is a global consensus on the need to reduce our collective carbon footprint. While much research attention has focused on developing alternative energy sources, automotive technologies or waste disposal techniques, we often ignore the fact that the ability to optimize (existing) operations to reduce their emissions impact is fundamental to this exercise. Business process management (BPM) technology, with its focus on understanding, modelling and improving/optimizing business processes, is a key starting point. Process modelling technology has applications beyond what we would traditionally describe as business processes - we can also model and improve manufacturing and other "physical" processes. We will use the term "Green BPM" to describe a novel class of technologies that leverage and extend existing BPM technology to enable process design, execution and monitoring in a manner informed by the carbon footprint of process designs and instances. This paper describes the first steps in the development of this class of technologies.

.A variety of international (Global Reporting Initiative 2009; United Nations 1998; The Green House Gas Protocol Initiative 2006; IPCC 2006), and national frameworks (Australian Government Department of Climate Change 2009a; US EPA 2009; European Commission 2004) exist to guide

and/or instruct organizations in this task. There is a strong interrelation between the frameworks, with national frameworks borrowing principles and ideas from various international frameworks. Throughout this paper we refer, if not otherwise stated, to the guidelines provided by (Australian Government Department of Climate Change 2009a), stating the six green house gases, whose emissions may be expressed in carbon dioxide equivalent (CO₂-e). These green house gases are emitted when performing a variety of activities, which are split into three scopes. Scope one (direct) emissions occur as a direct result of activities performed within an organization's boundary (e.g. combustion of fuel in vehicles or boilers). Scope two (indirect) emissions are externally emitted and are brought into the organizational boundary - in most cases electricity. Scope three emissions include all indirect emissions not part of scope two (e.g. emissions associated with material, flights by employees, etc.). Obligated organizations are required to account for the first two scopes, excluding scope three. With a worldwide growing number of obligated organizations software companies like Microsoft (2009), SAP (2009), Renison (2009), IBM (2009), Enablon (2009), Enviance (2009) and others identified this new market, providing software to account, report, model and even forecast an organizations carbon emissions. However, there is no existing approach that helps assess carbon emissions from a process perspective, providing deeper insights into the role of tasks and processes in an organization's total emissions and revealing opportunities for future carbon-aware process re-engineering. We begin by showing how to accumulate process annotations (in our case carbon emission annotations) through a process model for a selected task. Thereby we provide the answer to the question: 'How much carbon emissions would a process emit, if it would have been executed up to the selected point. This is not a trivial task providing a single answer since there might be various paths through a process model and during design time we do not know the one taken by a process instance. We then introduce a carbon model framework that can be used to populate a process diagram with emission annotation, providing an answer on the question: 'How much carbon emissions are emitted by executing a single task'. This is done by identifying all relevant carbon emitting entities and combining the principles of Bottom Up and Top-Down. In the Top-Down approach we leverage the principles of Activity Based Costing to find the contribution of each single task to the indirect carbon emissions. The idea to transfer ABC into the environmental domain was first mentioned by (Bras & Emblemsvag 2001) and is also used by (Renison 2009). In the Bottom-Up approach we use task specific information to determine its carbon emissions.

A variety of software solutions (SAP 2009), (Renison 2009), (IBM 2009), (Enablon 2009), (Enviance 2009) exist to support organizations in managing and optimizing their carbon emissions. However, no solution so far, is assessing carbon emissions during design-time from a process perspective, omitting significant optimization potentials in the area of business process reengineering.

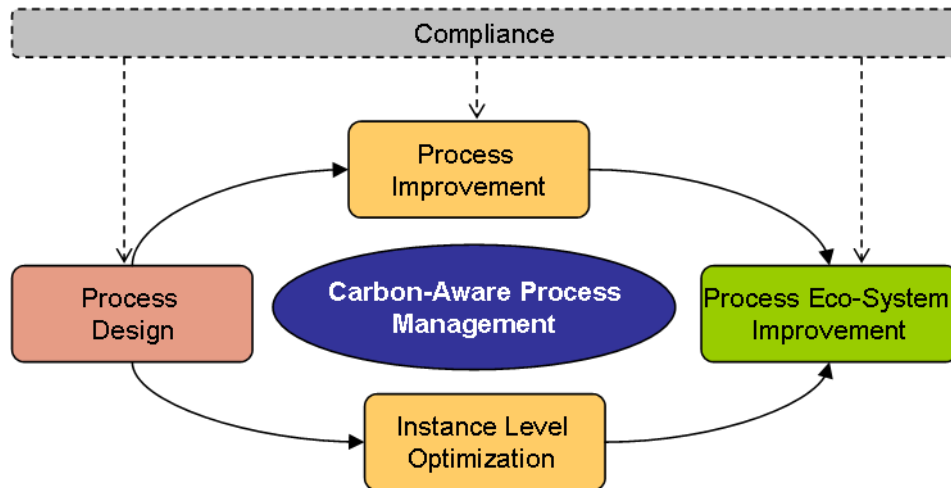


Figure 1: A framework for carbon-aware process management

Our work is based on the industry standard for business process modelling, the Business Process Modeling Notation (BPMN) (Object Management Group 2003). The idea to transfer cost accounting principles into the environmental domain is described by (Bras & Emblemstvag 2001) and also implemented by (Renison 2009). In terms of carbon emission related information used in our case study a variety of international (Global Reporting Initiative 2009; The Green House Gas Protocol Initiative 2006; IPCC 2006), and national frameworks (Australian Government Department of Climate Change 2009a; European Commission 2004) exist, with national frameworks mainly borrowing principles and ideas from various international frameworks. Through out this paper we refer, if not otherwise stated, to the guidelines provided by (Australian Government Department of Climate Change 2009a), stating the six green house gases, which's emissions are expressed in carbon dioxide equivalent (CO₂-e). These green house gases are emitted when performing a variety of activities, which are split into three scopes. Scope one (direct) emissions occur as a direct result of activities performed within an organization's boundary (e.g. combustion of fuel in vehicles or boilers). Scope two (indirect) emissions are externally emitted and are brought into the organizational boundary - in most cases electricity. Scope three emissions include all indirect emissions not part of scope two (e.g. emissions associated with material, flights by employees, etc.) Note that we do not focus on scope three emissions, since not enough information exists to usefully implement this scope and (Australian Government Department of Climate Change 2009a) and others do not require accounting for them.

DESIGN-TIME COMPUTATION OF CARBON EMISSION

Business process design must be informed (and driven) by the carbon footprint of the eventual process execution. Process design/modelling tools must support on-the-fly computation of the carbon footprint of a (possibly partial) process design, which can incrementally guide an analyst's choices throughout the design process. There are two key properties that the footprint evaluation of process designs must conform to:

- **Compositionality:** Compositionality requires that the cumulative carbon footprint of a process design must represent the composition of its constituent elements. The compositionality property also serves as a consistency constraint. Thus, the carbon footprint of a process model must be no less than the cumulative footprint of its constituent sub-processes and activities (loosely speaking - the precise accumulation procedure must make allowance for conditional splits, joins and loops).
- **Context-sensitivity:** Context-sensitivity is a property that carbon footprint measures used by any process design tool must satisfy. The energy footprint of most activities varies across geographies. The footprint can also vary depending on the time of day that a given activity is executed (based on impact on peak energy capacity). Any carbon footprint measure must therefore be annotated with the contextual assumptions that it is contingent on. Given the complexity of the contextual assumptions we envisage the use of a Process Emissions Ontology that would provide the vocabulary to describe the context on which a given carbon-footprint is contingent on (see Figure 2).

A process design tool must support two key functionalities. First, it must associate primary carbon footprint measures with elements of a process design (e.g. tasks, sub-processes, message flows) - these are externally obtained. Second, such a tool must compute secondary footprint measures by propagating primary measures (where available) through a process model.

Our objective is to identify the carbon emissions of a process at design time. Given any point in a process design we seek to understand the emissions that would have been generated if the process were to execute up to that point. This can be done by accumulating the carbon emissions of the tasks preceding the selected one. However, it is a not so trivial exercise, since during design time we do not know, for instance, which outgoing flow to follow after an XOR-, or OR-split. In other words there are various paths through a process model, resulting in more than one answer for the question being asked. Therefore we (1) seek to find all possible paths to a selected task within the process model, (2) determine the accumulated carbon emissions for selected task, for each path and (3) the average carbon emissions of a selected task.

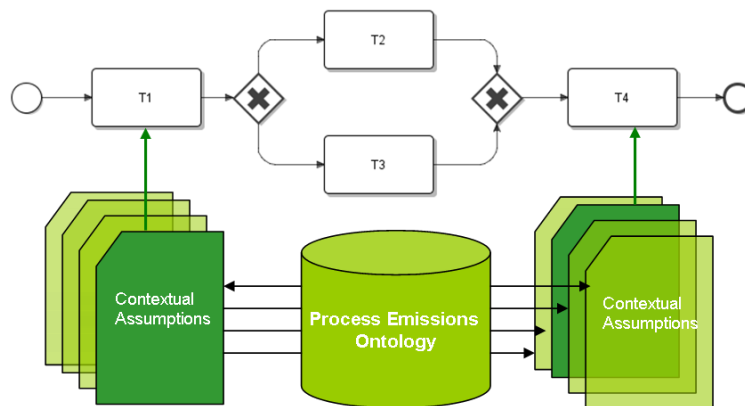


Figure 2: Associating carbon-footprints with process designs

TRAVERSING A PROCESS MODEL

We refer to the various paths from a start event to a selected task through a process model as scenario labels. A scenario label consists of a sequence ($\langle \rangle$) or a set ($\{ \}$) or a combination of both. Sets can be processed in any order and are used to consider parallel splits. The scenario labels for the selected task T7 in figure 6 are the following:

$\langle S1, T1, T2, \{ \langle T3 \rangle, \langle T4 \rangle \}, T5, T7 \rangle$.
 $\langle S1, T1, T6, T7 \rangle$.

We use the procedure described in (Hinge et al 2009) in the context of the ProcessSEER tool to compute the scenario labels describing alternative paths (from the Start Event) to the selected task.

ANNOTATING AND ACCUMULATING CARBON EMISSION

For the moment, we assume that the carbon emissions of each task within a process model are known and annotated, which we refer to as emission annotation. An emission annotation states the amount of carbon emissions, in terms of equivalent CO₂ amounts, a task emits when being executed. We refer to the accumulated emission annotations as emission scenario. The use of the word scenario is deliberate - each emissions scenario corresponds to an execution instance where the path defined by the associated scenario label is actually executed. An emission scenario at a given point in a process is the sum of (cumulative) carbon annotations that would have been emitted up to that point. We accumulate emission annotations step by step over a sequentially ordered pair of tasks following the sequence of the respective scenario label. The carbon emissions at a given point in a process design is thus a set of contingent measures, each corresponding to an alternative path from the Start Event to that point.

Let T_i and T_j be a pair of contiguous tasks connected by a control flow link. The set of (cumulative) effect scenarios at T_j consists of all distinct values $es(i)+ea(j)$, where $es(i)$ is an emissions scenario associated with T_i and $ea(j)$ is the emissions annotation of T_j . Each such distinct value constitutes an emissions scenario for T_j . We deal with AND-merges in the following manner. If T_i and T_j are the only two tasks immediately preceding an AND-merge, and T_m is the task immediately following it, the set of emissions scenarios at T_m consists of all distinct values $es(i)+es(j)+ea(m)$ for every distinct pair $es(i)$ and $es(j)$ such that $es(i)$ is an emissions scenario associated with T_i and $es(j)$ is an emissions scenario associated with T_j , while $ea(m)$ is the emissions annotation associated with T_m . In the preceding setting, if we replace the AND-merge with an XOR-merge, we proceed by $es(i') + ea(m)$ where $es(i')$ is an element of $es(i)$ or $es(j)$.

This procedure is sequentially repeated for all scenario labels. By sorting the scenario labels, according to their emission scenarios, we can provide insight about the most carbon friendly path, for a selected task, through a process model.

CARBON MODELLING FRAMEWORK

In the previous section we assumed that the carbon emissions of single tasks are annotated as emission annotations. In this section we introduce a framework for modelling carbon emissions

related entities and their relationship to determine the emissions of a single task. The focus is on the emissions associated with energy consumption, excluding emissions from industrial processes and waste management, since they require detailed knowledge about how the process is performed within each organization. However, to make these information available one can either start measuring them, to receive organization specific data, or pull more general data from a LCA database. In the following section we show how scope one and two entities can be modelled. Figure 3 summarizes our approach.

Dealing with Scope 2 Emissions

For scope two emissions we, for simplicity, will focus on electricity, which is in many cases the only contributor of scope two emissions. However, other scope two emissions for example heat and cooling can be modelled, using the same principles.

Emissions from electricity consumption are not activity specific. A unit of electricity consumed will always result in the same amount of carbon emissions no matter during which activity it was consumed. Since the green house gases are emitted outside the organizational borders the emission numbers are provided by the government for an electricity grid/region or by the electricity provider itself. Another scope two specific situation is that we are getting billed for our usage, resulting in information about the total consumption per period of time and enabling us to use a Top Down approach.

Scope 2 Top-Down Approach

Since we know the total consumption per period of time and the emissions are not activity specific we apply a top down approach, using the principles of cost accounting and eliminating the burden to determine all single electricity devices and their average consumption in an organization.

We use the principles of cost accounting to assigning costs, in our case electricity consumption, from a source, stating the total consumption, to shared resources and finally to tasks. A source can be populated by the electricity bill, or from an observed electricity meter. A shared source is an object consuming some part of the total electricity consumption, generally consisting of many single consuming devices. To accurately assign costs between shared resources, we use drivers, telling us the quantity of shared resources consumption. For example, we might want to allocate the total electricity consumption between our buildings (resources), by using the number of square meters per building as a driver. We can go further and allocate between resources, e.g. the different departments located in the building by using the number of workstations as a driver. Finally, the shared resources at the lowest level allocate their costs on all associated tasks. To determine the share of each task pulling from a shared resource, we might use the total amount of time a task is being performed by multiplying its average duration of execution, times the number of times executed and relate it to the other tasks pulling from this shared resource.

Scope 2 Bottom-Up Approach

In the top down approach we did not consider the consumption of single electrical devices used by a task. However, in some situations it makes sense to consider these devices, to which refer as atomic resources. Atomic resources are all relevant devices consuming a considerable amount of electricity, for which the average consumption is known, or worth determining. Thereby an atomic resource

states the average consumption for a service being provided and the tasks linked to it state the amount of service used. For example the atomic resource 'electric welder' states 2000W for the service welding and a task states to use the service for 5min per execution, resulting in a consumption of 167Wh per execution.

Considering the average consumption of scope two atomic resources, results in double counting of consumptions. To avoid this situation we subtract the total consumption per period of time of all tasks using an atomic resource from the total electricity consumption stated by the source. As a result, only the electricity consumption that is not directly associated with an atomic resource and its consumption is allocated between the shared resources and finally to the tasks.

In Figure 3 below we show the entities relationship as well as the flow of information between them.

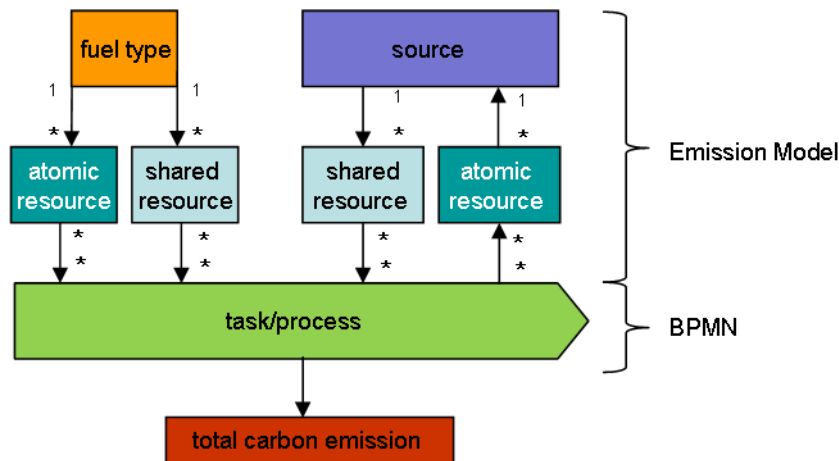


Figure 3: Modelling Scope 1 and 2 emissions

After applying the procedure we have the direct and indirect electricity consumption of each task being considered. By applying an emissions factor we can convert the consumed electricity in carbon dioxide equivalent emitted. Note that the same procedure can also be applied for heat and cooling.

Dealing with Scope 1 Emissions from Energy Consumption

Scope one emissions of energy consumption are emitted from the combustion of fuel types like gas, oil, coal, etc. within the organizational boundaries. The number of a green house gases emitted in CO₂ equivalent depends on the fuel type, the quantity used and how it is combusted. The figures and formulas to calculate the emissions are generally provided by the authority requiring accounting for the emissions. Note that we are referring to the Australian NGER Act (Australian Government Department of Climate Change 2009a) and that for other regulations might have different ones. However the principles are the same.

Since, the quantity of a fuel type and how it is combusted are activity specific, we have to apply a bottom up approach. Like before we use an atomic resource to state the average consumption for providing a service and additionally the fuel type being used. The task associated with the atomic resource specifies the intensity of the provided service used per execution. For example an atomic resource 'Truck' uses 30L of Diesel fuel to provide a distance of 100km. A task 'deliver goods' might use an average of 150km of that service, resulting in 45L Diesel combusted per execution. The carbon emissions from shared resources, being combustion devices that are used to generate for example heat, cooling or electricity (e.g. gas heater, generator, etc.) are allocated like scope two emissions with the exception that this time the organization is responsible for providing the necessary emission figures.

Note that scope one carbon emissions, not resulting from energy consumption, e.g. from chemical processes, gas and coal mining, or waste management can be annotated as well. As a result, we end up with annotated processes pulling emission data from the emission model and/or from external process emission databases or other sources.

CASE STUDY

In this section we show how the principles and procedures of the carbon-ware process design framework work together in a fictional case study.

Flex-Production Ltd. is a small tin-smith company in New South Wales, Australia. The company is not required by law to account for its carbon emissions, but the management decided to assess its emissions to reveal possible cost benefits.

The company has one facility with a Main building and a small Warehouse. The Warehouse, which is 500m away from the Main building, is used to store finished products. The Main building is separated into the units Production, where the tin-smith activities are performed and Back Office, responsible for finance, human resources and customer and supplier relationships. Every three months the company is receiving an electricity bill stating an average consumption of 280 kWh per week.

Assessing Atomic and Shared Resources

The company starts by identifying and assessing all relevant scope one and two atomic resources. It identifies the scope two resources Welder, Hammer and Cutter, which are all used for metal processing. Furthermore, the company identifies the scope one atomic resource Car, which is used to transfer finished products between the Main building and the Warehouse. The following information about the atomic resources are known:

AR(Welder): {2600W, Service Costs: 43Wh/ming}
AR(Hammer): {3000W, Service Costs: 50Wh/ming}
AR(Cutter): {2000W, Service Costs: 33Wh/ming}
AR(Car): {15L/100km, Service Costs: 0.15L/kmg}

In the next step the company identifies and evaluates all relevant shared resources, to allocate the total electricity consumption (costs) between them. After carefully consideration they decided to allocate the costs between the Main building and the Warehouse using the number of people as driver. The associated costs with the Main building are further allocated between the departments Back Office and Production, this time using the number of square meters as a driver. Note that other drivers could have been used, but the chosen ones best represent the relationship between the shared resources of this company.

SR(Main building): {driver: 9/10 employees - 90%}

SR(Warehouse): {driver: 1/10 employees - 10%}

SR(Production): {driver : 400/ 425 sqm - 94%}

SR(Back Office): {driver : 25/425 sqm - 6%}

Assessing Business Processes Having all atomic and shared resources identified, the company is assessing its processes. Due to space limitations we only consider in full detail one process, being performed in the department Production.

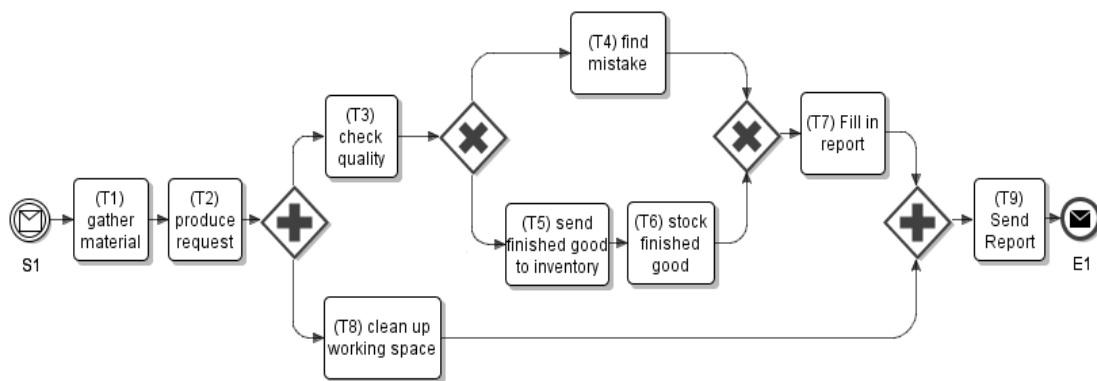


Figure 4: Example business process

Process Description

After receiving a production request, the workers gather the required material and start producing the request (see Figure 4). After the production is finished a supervisor checks the product quality, while other workers, due to safety regulations, clean up the working space. If the supervisor finds a mistake he/she fills in a report and sends it to the Back Office. If the product quality is acceptable it is sent by car to the warehouse, where it is stocked by the responsible worker and a report, informing the Back Office about the completion is send.

The tasks of the process are annotated with duration of a task (d), number of times executed (n), shared resource used (SR), atomic resource used (AR):

- gather material: {d=1 min, n=154/week, SR=[Production]},
- produce request: {d = 30 min, n = 100/week, SR = [Production], AR =[Welder, 10min], AR = [Hammer, 10min], AR = [Cutter, 15min]},

- check quality: {d = 10 min, n = 140/week, SR = [Production]},
- clean up: {d = 12 min, n = 122/week, SR = [Production]},
- find mistake: {d = 17 min, n = 45/week, SR = [Production]},
- send finished good to inventory: {d = 10 min, n = 91/week, AR = [Car, 1km], SR = []},
- stock finished good: {d = 5 min, n = 122/week, SR = [Warehouse]},
- fill in report: {d = 2 min, n = 153/week, SR = [Production]},
- send report: {d = 1 min, n = 254/week, SR = [Production]},

One can see that all tasks except of 'send finished good to inventory' and 'stock finished good' (executed in the Warehouse) are executed in the domain of the Production department in the Main building. The electricity consumption of the Production department is only allocated between the tasks in its domain, which we show later.

Finding Scope 2 Carbon Emissions

In this section we explain how the carbon-aware process design framework deals with the given figures and relationships.

Applying Bottom Up Approach

In total the company identified three electrical atomic resources, which are used in the task 'produce request'. The tasks annotation states that the atomic resource hammer is used 10min resulting in 500Wh consumed per execution. The Welder is used 10min, resulting in 430Wh consumed, and the atomic resource Cutter is used 15min resulting in 490Wh consumed per execution. The task 'produce request' is executed 100 times per week, which results in the total electricity consumption of atomic resources peer week (500Wh + 430Wh + 490Wh) 100 = 142k Wh. This figure is subtracted from the total amount of electricity used per week (280 kWh), resulting in 138kWh to be allocated between the shared resources.

Applying Top Down Approach

The company has identified four shared resources, viz. Main building, Warehouse, Production, Back Office. Using the already determined drivers the 138 kWh are allocated, whereby (138 times 0.9) 111.78 kWh fall on the Main building, (138 times 0.1) 13.8 kWh on the Warehouse, (111.78 times 0.94) 105.1 kWh on Production department, and (111.78 times 0.06) 6.7 kWh on the Back Office.

To determine the responsibility of single tasks being in the domain of a shared resource the total amount of time a task is executed per week is used as driver. In other words the more and longer a task is executed the higher its share of the costs. The tasks being executed in the domain of Production, their share and electricity consumption per execution and resulting emissions are:

- gather material (154min/week - 2.1% - 2.21kWh/week - 14.4Wh/ex),
- produce request(3000min/week - 40.86% - 42.94kWh/week - 429.Wh/ex),
- check quality (1400min/week - 19.07% - 20.04kWh/week - 143.1Wh/ex),
- clean up (1464min/week - 19.94% - 20.69kWh/week - 169.6Wh/ex),

- Find mistake (765min/week – 10.42% - 10.95kWh/week – 28.6Wh/ex),
- Fill in report (306min/week – 4.17% - 4.38kWh/week – 28.6Wh/ex),
- Send report (254min/week – 3.46% - 3.64kWh/week – 14.3Wh/ex).

In the domain of the Warehouse, besides 'Stock Finished Good', several other tasks are being performed (due to space limitations we do not describe these tasks in detail) with a total duration of 2000 min per week:

- Stock finished Good (610min/week – 30.5% - 4.21kWh/week – 34.6Wh/ex)

CALCULATING CARBON EMISSIONS

Combining the electricity consumption from shared resources with the consumption from atomic resources we can calculate the scope two emissions, using the emission factor (0.89CO₂-e/kWh for New South Wales) provided by (Australian Government Department of Climate Change 2009b):

- gather material(14.4Wh/ex – 0.01 kgCO₂e),
- produce request (429.4Wh/ex+ 1420Wh - 1.65 kgCO₂e),
- check quality (143.1Wh/ex - 0.13 kgCO₂e),
- clean up (169.6Wh/ex - 0.15 kgCO₂e),
- find mistake (243.3Wh/ex - 0.22 kgCO₂e),
- fill in report (28.6Wh/ex - 0.03 kgCO₂e),
- send report (14.3Wh/ex - 0.01 kgCO₂e),
- stock finished good (34.6Wh/ex - 0.03 kgCO₂e).

Scope 1 Carbon Emissions

The company identified a single car as a scope one atomic resource, stating an average consumption of 15L=100km. The car is used by the task 'send finished good to inventory', with an intensity of 1km, resulting in 0.15L Gasoline fuel combusted per execution. Applying the energy content and emissions factor for transportation activities, we calculate the following carbon emissions from fuel combustion:

- send finished good to inventory (0.15L/ex – 0.371 kgCO₂e/ex¹⁹)

Design Time Computation of Carbon Emissions

After applying the procedures we end up with the following carbon emissions associated with the tasks gather material (0.01 kg CO₂-e), produce request (1.65 kg CO₂-e), check quality (0.13 kg CO₂e), clean up (0.15 kg CO₂e), find mistake (0.22 kg CO₂e), fill in report (0.03 kg CO₂e), send

¹⁹ 0.342 kg CO₂e from CO₂, 0.023 kg CO₂e from N₂O, 0.006 kg CO₂e from CH₄

report (0.01 kg CO₂e), stock finished good (0.03 kg CO₂e) and send finished good to inventory (0.37 kg CO₂e). Note that if we want to make an additional emission annotation using a database or another source, we can do so but have to consider the annotation when summing up the emission for a task.

PROCESS IMPROVEMENT

Building on machinery for assessing the carbon footprint of process designs, we must address the question of process improvement for carbon footprint optimization. The focus here is on design improvement - we will address instance-level optimization later. Process improvement must therefore involve process re-design to obtain processes that achieve the same (functional) goals, while minimizing the carbon footprint (and potentially other non-functional criteria as well). To achieve this, we need: (1) the ability to annotate process designs with detailed specifications of functional goals, (2) the ability to assess proximity of process designs to other process designs and (3) the ability to search for optimal designs through a space of alternative process designs. Figure 5 summarizes the framework.

The ProcessSEER system (Hinge et al 2009), building on our earlier work in (Ghose et al 2007), (Koliadis et al 2007), provides the ability for analysts to specify context-independent *immediate effects* of tasks (in a user-accessible controlled natural language, with underlying formal translations), which are then *propagated* and *contextualized* to obtain cumulative effect annotations for each task. The resulting semantic effect annotation of process designs is critical in ensuring that process functionality is not negatively impacted during a process improvement exercise, either at the level of an entire process or at sub-process level. Process improvement can be viewed as a state space search problem, in state space composed of alternative process designs (each of which achieves the functionality of the original process), with the cumulative carbon footprint of a process being the primary indicator of the “quality” of a solution. While the space of alternative designs if often large, it is nonetheless possible to devise automated machinery that performs search for the optimal process (re-)design – given that process improvement is an off-line exercise and there is no obligation to provide real-time solutions.

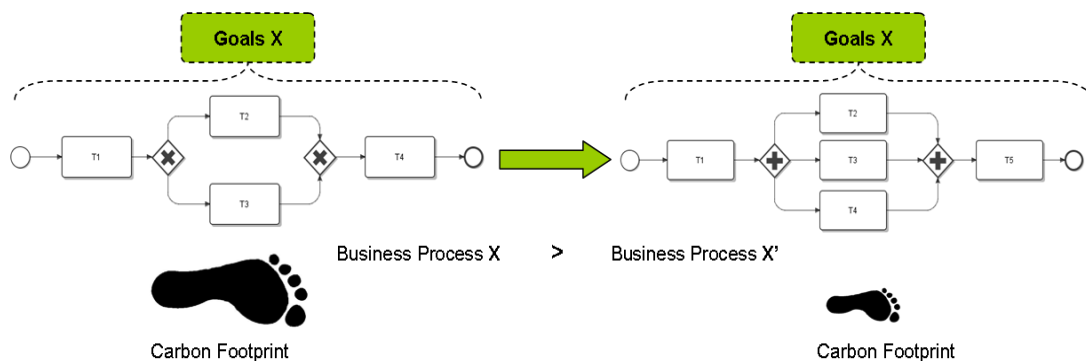


Figure 5: Green process improvement

OPTIMIZING PROCESS ECOLOGIES

Most enterprise, and cross-enterprise, contexts involve large networks of inter-operating processes. Changes to a given process often impact other processes, leading to a view of *process ecologies* composed of inter-operating processes. Process improvement often faces the local vs. global optimization dilemma. Local optimization, at an individual process design level, is easy to achieve – yet a set of locally optimized processes is not optimal if one takes a global view. The state space search based machinery for process optimization described in the previous section needs to be extended *globally* optimize sets of inter-operating processes. Figure 6 summarizes this view.

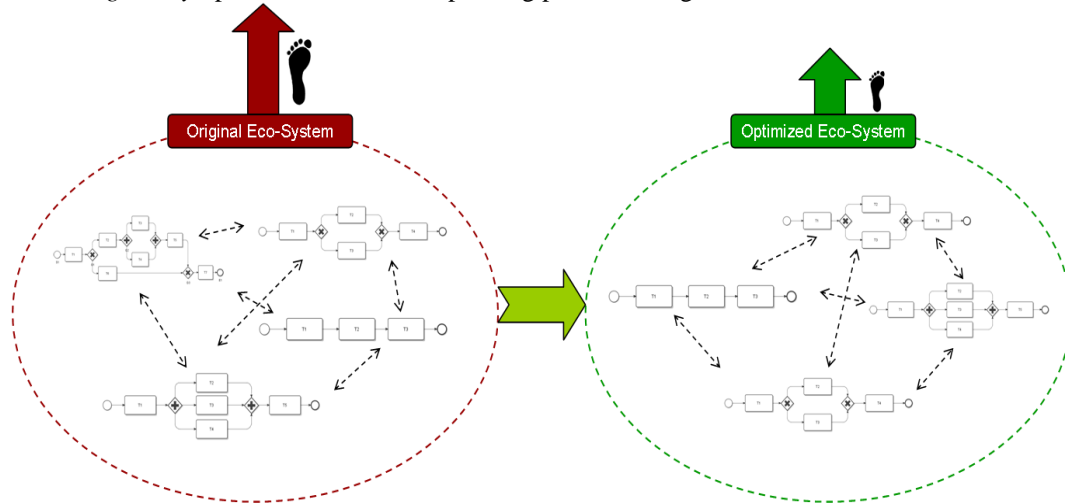


Figure 6: Optimizing process ecologies

INSTANCE-LEVEL PROCESS OPTIMIZATION

While much of the preceding discussion focuses on process designs, process execution management can also contribute to carbon footprint minimization, in several ways. First, monitoring process execution can ensure conformance with carbon-optimized process designs. Where the infrastructure required for monitoring process execution is expensive (as is often the case), activity triage (i.e., the prioritization of activities within processes) can help direct investment in monitoring infrastructure towards those elements of a process that have the most impact on the carbon footprint. Second, process mining can help generate "as-is" process models (van der Aalst et al 2004) which can help uncover situations where process executions deviate on a fairly routine basis from optimized process designs. Third, process engines can be instrumented to exploit opportunistic optimization opportunities during execution. Consider the typical pre-departure preparation process for aircraft at airports with sub-zero temperatures, which involves refuelling, de-icing and take-off, in that sequence. The critical constraint to satisfy is one that specifies the maximum time that may elapse between de-icing and take-off (say t minutes), and is the key reason for adopting the refuel-deice-takeoff sequence. In other words, an alternative permutation of these activities is in principle

permissible, provided the de-ice-to-takeoff constraint is satisfied. Consider an instance of this process for a specific aircraft, which is obliged to idle, loaded, at the departure gate due to a delayed fuel truck. If there was a guaranteed arrival time for the fuel truck, and the sum of the estimated refuelling time and waiting time for take-off were well under the maximum de-ice-to-takeoff interval of t minutes (while still allowing a safety margin), a possible optimization is to de-ice, then refuel and take-off, only for that process instance. Any machinery for leveraging such opportunistic instance-level optimizations must be equipped with the hard constraints associated with a given process design (such as the de-ice-to-takeoff interval), as well as sophisticated temporal reasoning capabilities, as the example above illustrates.

PROCESS COMPLIANCE

With increasingly onerous legislative and regulatory environmental obligations that business processes must satisfy, compliance management has emerged as a critical component of green BPM. There are two key aspects of compliance management: (1) compliance checking and (2) non-compliance resolution. Both exercises could be conducted in the context of process designs or process instances. Design-time compliance checking requires us to establish that process designs do not violate compliance requirements while run-time compliance checking requires us to establish the same for process instances.

CONCLUSION

We have provided a roadmap for research and development in this nascent area of carbon-aware “green” business process management. We have provided a detailed description of the machinery for assessing the carbon-footprint of process designs, and outlined how this might be leveraged in process improvement, both at the single process level and at the level of networks of inter-operating processes. There is much to be done, and this paper should serve as a call to arms.

REFERENCES

- Australian Government Department of Climate Change (2009a) “About the national greenhouse and energy reporting act”, <http://www.climatechange.gov.au/reporting/>. last checked 10.09.2009.
- Australian Government Department of Climate Change (2009b) “National green house accounts (nga) factors”, <http://www.climatechange.gov.au/workbook/pubs/workbook-jun09.pdf>, last checked 10.09.2009.
- Bras B. and Emblemvag J. (2001) “Activity-Based Cost and Environmental Management”, *Massachusetts: Kluwer Academic Publishers*, 2001.
- Enablon (2009) “Enablon ghg-ms”, <http://enablon.com/products/corporate-responsibilityehs-management/GHG-emissions-management/functionalities.aspx>, last checked 10.09.2009.
- Enviance (2009) “Solutions”, <http://www.enviance.com/solutions/compliancemanagement-software.aspx>, last checked 10.09.2009.
- European Commission (2004) “Monitoring and reporting of greenhouse gas emissions under the emission trading directive”, <http://eurlex.europa.eu/LexUriServ/site/en/oj/2004/l059/105920040226en00010074.pdf>, last checked 10.09.2009.

- Ghose A. and Koliadis G. (2007) "Auditing business process compliance", in *Proceedings of the International Conference on Service-Oriented Computing (ICSOC-2007)*. Global Reporting Initiative (2009) "Sustainability reporting guidelines", <http://www.globalreporting.org/>, last checked 09.09.2009
- Hinge K., Ghose A., and Koliadis G. (2009) "Process seer: A tool for semantic effect annotation of business process models", in *Proc. of the 13th IEEE International EDOC Conference (EDOC-2009)*, I.C.S. Press, Ed., 2009.
- IBM (2009) "Ilog logicnet plus carbon footprint extension", <http://www.ilog.com/products/logicnet-plus-xe/carbon-footprint/>, last checked 10.09.2009.
- IPCC (2006) "Ipc guidelines for national greenhouse gas inventories" <http://www.ipccngip.iges.or.jp/public/2006gl/index.html>, last checked 09.09.2009.
- Koliadis G. and Ghose A. (2007) "Semantic verification of interoperational business process models", in *Proceedings of the 2007 IEEE Services Computing Conference (SCC-2007)*.
- Microsoft (2009) "Microsoft helps businesses manage their carbon footprint and identify cost-saving opportunities", <http://www.microsoft.com/presspass/press/2009/feb09/02-09JDTPR.mspx>, last checked 09.09.2009.
- Object Management Group (2003) "Business process modeling notation (bpmn)", <http://www.omg.org/docs/formal/0901-03.pdf>. last checked 10.09.2009.
- Renison K. (2009) "Carbon footprint modeling with sas for sustainability management: Beyond calculation", <http://support.sas.com/resources/papers/proceedings09/343-2009.pdf>. last checked 09.09.2009.
- SAP (2009) "Emissions management with sap environmental compliance", <http://www50.sap.com/businessmaps/6F0CD48E4C44451C8E0CAB0FD3365716.htm>, last checked 10.09.2009.
- The Green House Gas Protocol Initiative (2006) "Programs and registries", <http://www.ipcc-nggip.iges.or.jp/public/2006gl/index.html>, last checked 09.09.2009.
- US EPA (2009) "Ghg inventory guidance", <http://www.epa.gov/stateply/resources/inventoryguidance.html>, last checked 10.09.2009.
- United Nations (1998) "Kyoto protocol to the united nations framework convention on climate change", <http://www.ghgprotocol.org/programs-and-registries>, last checked 09.09.2009.
- van der Aalst W., Weijters T., Maruster L. (2004) "Workflow Mining: Discovering Process Models from Event Logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128-1142.