Integrating Sustainability into Computer Science Education: A Survey and Framework

CompSustNet Technical Report 1

Douglas H. Fisher | Zimei Bian | Selina Chen

Contact:
douglas.h.fisher@vanderbilt.edu

ABSTRACT

This paper surveys efforts at the integration of sustainability into the computing curriculum at three levels of granularity, with our particular focus on higher education. **Course-level integration** introduces whole courses into a computing curriculum that are focused on topics at the intersection of computing and sustainability. At finer granularity is **component-level integration**, which introduces sustainability-topical exercises, assessments, lectures, modules, and other materials into computing courses where the dominant focus is on the learning of computing concepts, techniques, and applications. Finally, we draft a possible future of **curricular-level integration** of sustainability into computing education, which can include special tracks and majors.

Keywords

Computational sustainability, sustainability, resource repositories

1. INTRODUCTION

Computing is ubiquitous, and potentially benefits and impinges on many environmental and societal sustainability issues. It is no surprise then that computing is included across sustainability-relevant education and research. Courses in agent-based and other simulation modeling for environmental applications are good examples of the penetration of computing into sustainability-topical curricula. For example, consider Environmental Science 475 from Southern Oregon University (SOUES475, ongoing):

> **ES 475 Environmental Modeling** This course teaches environmental simulation modeling and the application of computer model results to real world problems in environmental studies. Computer-driven data analysis and modeling have become critical to the understanding and resolution of environmental problems and issues of sustainability and natural resource management. This course teaches computer simulation modeling skills and environmental system dynamics through an applied approach requiring the conceptualization, construction and creation of dynamic computer simulation models to aid in the resolution of environmental problems. (SOUES475, ongoing)

There are important computing concepts and techniques presented in such courses, which can be a valuable source of ideas for teaching computing generally, including within a computer science curriculum.

Inversely, sustainability issues can be infused into the curricula of computing; this directionality is the focus of our paper. The reasons advocated for this infusion of sustainability into computing are several fold (CCC/CRA, 2011, pp. 20-21; NAP, 2012, Ch. 5, pp. 96-97).

First, sustainability concerns are (or should be) pervasive, as are the disruptive (aka potentially transformative) technologies of computing. For the benefit of society and environment, a sustainability agenda and a computing agenda should be coordinated, if not coextensive.

Second, computer science educators better ensure that computing is "taught right" in sustainability fields, and other application fields for that matter, if CS educators embrace sustainability applications and lead on developing pedagogy and materials for computing technologies in these areas, ideally in collaboration with sustainability-area domain experts.

Third, sustainability areas offer rich real-world problems that can be crafted at varying complexity, all of which can benefit the education of computer science students.

This paper surveys efforts at the integration of sustainability into the computing curriculum at three levels of granularity, with our particular focus on **higher education**. The space of computing and sustainability education and research has been termed **computational sustainability** (Gomes, 2009), a term that we adopt here, but we also recognize that much work in computing and sustainability is not labeled as such (e.g., Mankoff, et al, 2008), and we recognize relevant work regardless of label.

Course-level integration introduces whole courses that are focused on computational sustainability topics. At finer granularity is **component-level integration**, which introduces sustainability-topical exercises, assessments, lectures and other materials into courses where the dominant focus is on learning of computing concepts, techniques, and applications. These types of integration are proposed elsewhere too (Cui, 2010; Fisher, 2007), though an actual survey of available courses and materials is new.

Finally, we draft a possible future for **curricular-level integration** of sustainability into computing education. While we don't imagine creation of computational sustainability majors in the near term, many universities allow for student-customized majors (Fehlen, 2016), of which computational sustainability might be an attractive option.

Our motivations for surveying and framing the integration of sustainability into computing curricula are (a) to provide resources for instructors interested in creating their own courses, including wiki resources for adding to the community's resources; and (b) to provide thoughts on computational sustainability curricula that instructors may find beneficial for advising students.

2. Course-Level Integration

Perhaps the most apparent type of integration of sustainability into computing education is the introduction of specialized computational sustainability courses. A Google search using a core set of keywords (i.e., 'sustainability computer science course'), with variations that include additional keywords, and quotes, turns up a number of such courses offered at diverse institutions of higher education. But our search for relevant coursework has been ongoing for a longer time. Indeed, one of our primary motivations in this paper is to catalog computational sustainability courses and materials, and to highlight the crowdsourcing mechanisms that are already created, which others can use to add to online repositories. The References include links to the courses and materials cited herein.

In this section and throughout the paper, we will focus attention on integration of sustainability into computing curricula. Having noted the relevance and importance of the inverse direction, to the extent it can be distinguished, of computing into sustainability, we set it aside in this paper, other than briefly returning to it in Section 4. Thus, some courses in the intersection of computing and sustainability have been excluded here because their home department is not computing primary -- this includes the exclusion of courses in non-computing primary engineering disciplines.

2.1 Green IT Courses

Some courses are concerned exclusively with what is popularly called "Green IT." It is widely recognized that there are sustainability concerns with information technology itself, most notably computing. Köhler & Erdmann (2004) reference these as the "first order effects" of information technology. The first order (aka direct) effects include the growing energy footprint of computing worldwide, and thus there is an interest in reducing per "computing unit" energy usage, as well as the energy used to create and recycle computing hardware. Other concerns relate to the depletion of precious metals and other materials used in hardware, and the difficulty with recycling these materials.

At the level of topic abstraction in which we will subsequently consider other computational sustainability courses, those courses that are predominantly "Green IT" (e.g., GMUCS795, 2016; BUCS504, 2012; Hamilton, 2015; Cui, 2010) form their own category. A further decomposition of Green IT courses would

* characterize the extent to which the focus was on energy, to the exclusion of materials depletion and recycling (GMUCS795 2016; Hamilton, 2015), or in combination with materials depletion and recycling (BUCS504, 2012);

* characterize the course's attention to the entire technology lifecycle (BUCS504, 2012), versus just the usage phase (GMUCS795 2016; Hamilton, 2015); and

* characterize the kinds of technology considered, to include mobile devices, networks, the cloud and server farms, and virtualization.

Finally, a deeper characterization of Green IT courses would ask whether they have any coverage of "rebound effects" (Fisher, 2012; Tomlinson, et al 2011), which are indirect effects (e.g., increased collective worldwide energy footprint) that result from changes in technology (e.g., greater per computing-unit energy efficiency) for a variety of reasons (e.g., "induced use", in which increased efficiency encourages greater use). The Green IT courses we surveyed have no such coverage. Indeed, such coverage would imply an attention to larger system-level considerations about how humans use technology, to include HCI, and by the definitions we have given here, such coverage would make the course NOT exclusively Green IT.

Other courses that we survey next may include some Green IT as part of their content, but they are broader in scope, addressing what Köhler & Erdmann (2004) call second and third order effects of computing. Tomlinson (2010) refers to such higher-order effects as "*Greening through IT*". These higher order effects will typically relate in one way or another to computing's importance in intelligent, evidence-based decision making about environmental and societal challenges. This is much more in line with what Gomes (2009) and others have intended by the term computational sustainability.

2.2 Computational Sustainability Courses

Of the computational sustainability courses that we have found, some appear to have no public facing presence beyond brief catalog course descriptions (SewaneeCSCI120, ongoing; PittCS690, ongoing; NDSUCSCI428, ongoing). Nonetheless, having been found, instructors will be invited to contribute details.

Other courses, such as two current Fall 2016 offerings (UNMARTS441/541, 2016; VUCS3892/5892, 2016), have much more publicly available material than the catalog description. For these latter courses, the available online material includes required and optional readings, and assignment and project deadlines. The course description for the University of New Mexico course, for example, is here, and a link to the complete syllabus is found under the References (which is the case for all courses, when those resources are available).

> **ARTS 441/541, ECE, CS, SUST, CRP: Computational Sustainability**, Computational sustainability focuses on computational methods for balancing environmental, economic, and societal needs for a sustainable future. It is a highly interdisciplinary field full of diverse developments. The course is designed to be an introduction to computational sustainability, providing a broad coverage of the field. It is suitable for advanced undergraduate and graduate students in computer science, computer engineering or from other disciplines with good familiarity with computational methods. (UNMARTS441/541, 2016).

2.2.1 Computational Sustainability Course Topics

To better characterize and understand the coverage of topics in computational sustainability courses to date, we used the Mallet topic modeling software (Mallet, 2016) from UMass Amherst to discover plausible topic models from the readings of 6 previously offered courses. These previously offered courses are:

Sustainability and Assistive Computing (Bryn Mawr College, Fall 2010);

Computing and the Environment (Vanderbilt University, Spring 2011);

Topics in Computational Sustainability (Cornell University, Spring 2011);

Computational Sustainability (University of British Columbia, Winter 2013–2014);

Computational Sustainability (Georgia Tech, Spring 2014);

Seminar on Computational Sustainability: Algorithms for Ecology and Conservation (University of Massachusetts Amherst, Spring 2014).

experimenting with paper abstracts (e.g., replacing each instance of "models" with "model"), we have not yet done this for full papers, so "word" and "words" both appear in topic 5's keyword list, and to simply remove 'models', for example, would artificially diminish the role of modeling in the definition of topic 5 and potentially other topics.

Tomlinson (2010) was a required textbook in the Vanderbilt course. This text largely focused on social computing for sustainability, but it was not online, and probably would not have been used in any case given its disproportionate length. Thus, social computing is probably underrepresented in the topic list.

TOPICS GENERATED			
Topic #	*Weight*	*Keywords*	*Topic Name*
0	0.15074	energy power data consumption time carbon electricity environmental system	GreenIT/Energy
1	0.18246	problem algorithm set time sensor greedy network number optimal	Optimization/Sensor
2	0.16311	data environmental urban energy services development science land government	Urban/Policy
3	0.09139	problem cost solution budget corridor connectivity habitat connected conservation	Optimization/Land
4	0.08485	waste electronic media hazardous equipment social nigeria computer countries	GreenIT/Materials
5	0.27841	model data models species distribution set maxent detection modeling	Modeling/Species
6	0.11874	energy building cost design optimization model optimisation objective buildings	Optimization/Built
7	0.09318	model capture data survival time models rates parameters recapture	Modeling/Method
8	0.12163	food network species webs web time information data networks	Ecology Webs
9	0.09067	climate change global water ocean sea earth fish system	Earth Systems

Table 1: *Ten topics discovered with Mallet when run on full papers of all readings of 6 courses. Each topic is defined by 9 keywords. Each topic is weighted to reflect its prevalence in the document collection. Because a document can reflect multiple topics, these weights do not sum to 1.0. The topic name was selected by the authors, as is typical when using topic modeling.*

The materials for these courses, and others, are in a Wikibook repository that is open to community readership, authorship, and editorship (Wikibook CompSust courses, 2016 -- see References for the link). It is to this repository that instructors can add details of their computational sustainability courses. We plan to update and refine the kind of initial analysis reported here at intermittent points, and informed by community feedback. While this initial cohort of courses is small enough at present, and our domain expertise is good enough, that topic modeling is not yet required, we imagine a future in which cataloging of subsequent materials is aided through topic modeling (e.g., Ecowatch, ongoing).

Given a set of documents, Mallet returns a set of "topics" that are represented in the collection. For example, given a set of scientific papers from multiple academic disciplines, we might expect that Mallet would "rediscover" the disciplines from which the documents were drawn. Mallet uses word prevalence, co-occurrences, and other factors to guide the definition of topics. The topics themselves are characterized by a list of dominant words that tend to be found in the documents of that topic.

To use Mallet, we took all electronically available readings, which included virtually all of the readings from across all 6 courses above, and treated this union as a single document collection. The topics discovered by Mallet are in Table 1.

Using Mallet, and other topic modeling packages, for the discovery of topics is almost never a fully automated process. Rather, analysts -- the authors in this case -- ran Mallet with a variety of parameter settings (e.g., the target number of topics was varied) and we initially experimented with simply using the abstracts of readings rather than the full papers. While we had done manual stemming when

With the discovered topics in Table 1, Mallet is able to characterize a given document by a weighted vector of topics. In particular, after topic discovery, we appended the readings for each course into its own "super-document" and ran Mallet's characterization procedures on each course's super-document, which we take as a topic characterization for the course. The results of characterizing each course by a weighted topic vector are shown in Table 2. We will continue experiments post review.

In Table 2, each course is represented by 10 topic weights, representing the degree that the course's readings reflect each topic. For a given course, the ten weights sum to 1.0. Thus, the average topic weight is 1.0/10 or 0.1. We take each topic with a weight that is greater than this average to be "characteristic" of the course. Thus, the characteristic topics of the Bryn Mawr course are Topics 1, 2, 4, and 5. We use bold face and font size to for visualizing the characteristic topics across courses. The same caveats apply as did in topic discovery (e.g., Vanderbilt's use of a social computing textbook is not reflected in its characterization).

Despite the imperfect precision of topic modeling for characterizing our course collection, we believe that refining the methodology is worth pursuing, thereby allowing us to track changes in computational sustainability course coverage over time. For example, as an additional analysis, we want to separate discovery of computing topics from discovery of sustainability topics. Currently, these two areas are conflated in one topic modeling analysis, which is interesting, but it may result in sustainability topics suppressing computing topics. Indeed, with the exception of topic 7 in Table 1, there seems to be a bias towards sustainability-topical keywords.

Even in its current preliminary form, the characterization using topic modeling enables interested instructors and learners to see what topics appear most common across courses (e.g., 1, 2, 3, 5) and which topics are somewhat unique to a given course (e.g., 4, 7, 9). In many cases this information can be useful for planning a syllabus that covers important bases and fills gaps.

3. Component-Level Integration

Finer grained integration of sustainability can happen within the substrate of an individual course. For example, Abernethy & Treu (2014) introduce a week-long module into an introductory ("familiarity") computing course, as well as a module into an upper-division project management course.

COURSE TOPIC WEIGHTS					
School	*Topic 0*	*Topic 1*	*Topic 2*	*Topic 3*	*Topic 4*
Bryn Mawr	0.090943549	**0.127644406**	**0.20480037**	2.10E-05	**0.265664737**
Cornell	7.22E-05	0.085409982	**0.174295598**	0.009161242	0.005980967
Georgia Tech	0.081458989	**0.136824135**	**0.100419814**	**0.125061275**	0.061678773
UBC	**0.200559536**	0.018010526	**0.172902203**	0.044725581	0.052835175
UMass Amherst	1.87E-05	**0.177675797**	6.20E-04	**0.217023506**	2.66E-06
Vanderbilt	**0.354199272**	0.033780717	0.02020729	**0.253033232**	0.072572848
School	*Topic 5*	*Topic 6*	*Topic 7*	*Topic 8*	*Topic 9*
Bryn Mawr	**0.29306572**	0.001092996	0.002332577	0.005188805	0.009245879
Cornell	0.054950987	0.056984767	0.089727397	**0.474219654**	0.04919718
Georgia Tech	**0.193939583**	**0.14640088**	0.028616956	0.038639172	0.086960423
UBC	**0.102387938**	**0.100914674**	5.24E-05	0.010594252	**0.297017732**
UMass Amherst	**0.284061303**	0.030038263	**0.283903305**	0.006486598	1.70E-04
Vanderbilt	0.048782513	0.020952409	2.51E-04	**0.137485102**	0.058735835

Table 2: *Topic representation in each of 6 courses. See main text for explanation.*

At even finer levels of granularity, infusion can happen through the introduction of video lectures, readings, exercises, and projects. Rather than being component materials for a computing and sustainability topical course, these fine-grained infusions happen in computing courses in which the almost-exclusive focus is on teaching and learning of computing concepts.

Often, sustainability-focused introductions are particular to an instructor or class, even if they are sometimes available on the Web. For example, a project used in an introductory database management course (VUCS3265, ongoing) has students reverse engineer the database used in support of Oberlin's dorm energy monitoring initiative (Petersen, et al, 2007), and in the process, students learn about the environmental consequences of technology. However, there is undoubtedly a magnifier effect when sustainability materials make their way into more widely disseminated textbooks, such as Patterson and Hennessy's (2005) computer organization textbook, in which real-world applications, most with considerable sustainability relevance, are interleaved between the regular technical chapters.

3.1 Computational-Sustainability Repositories

Instructors can contribute to and use materials from resource repositories that are specific to computational sustainability. For example, the SISL (Sustainability Improves Student Learning) project includes pages of suggested educational resources for incorporating sustainability into many disciplines, including in computer science (SISL CS, lapsed -- See references for URL). Unfortunately, funding for the SISL-sponsored pages, including the SIGCSE-cosponsored CS page, has lapsed (SISL, personal communication), and while updates by the SISL CS author are theoretically ok, such updates are fragile, with any breaks introduced through editing not being correctable on the SISL side.

While SISL CS (lapsed) is computing wide, albeit under developed, it can be copied, expanded, and maintained by the computing community on another site. A model for community stewardship is on Wikimedia, such as the sustainability "lab companion" for standard courses in artificial intelligence (Wikibooks, AI Lab). The goals of this site are to be a repository for sustainability-focused materials that any AI instructor can use -- first and foremost, to illustrate and exercise AI concepts and techniques.

The ultimate success of these niche repositories may be helped by lessons learned, and perhaps by piggybacking on repositories of a more general nature.

3.2 Integration into General Repositories

Nifty assignments (Parlante, ongoing) is an established and vetted repository of exercises and projects for computing generally. Along with repositories that are more general still (e.g., IEEE Real World, ongoing; Merlot, ongoing), Nifty Assignments is another target for and source of sustainability-relevant CS educational materials.

A survey of Nifty assignments doesn't show any exercises that are substantially sustainability primary, but some are clearly close and could be adapted to a sustainability-topical problem. For example, Wator World (Scott, 2011) points to relevant biology background that could be highlighted and perhaps expanded, so that there is infusion of yet more sustainability-relevant content, while not detracting from the CS concepts. Another example is Mountain Paths (Franke, 2016), for finding paths through mountainous terrain. If paired with background such as Brunskill & Lesh's (2010) accessible write-up on finding optimal paths for rural, developing world healthcare workers, who walk many miles a day, then varying adaptations of the exercise could be compelling sustainability-relevant exercises.

If executing a strategy for adapting existing exercises, or simply using exercises as is, then ideally those exercises can and should be cited, as we have done in this section. Citation better ensures that the contributions of educational material will be recognized, and that its usage will tracked through analytics on publicly accessible platforms such as Google. It is worth community effort to decide upon conventions for citation, which can then be posted to repositories.

4. Curriculum-Level Integration

A third, coarsest proposed level of granularity for integrating sustainability into computing education is through the introduction of curricular-level constructs like minors, tracks, and majors. This section is largely speculative.

Here we focus on possible computational sustainability majors, in part because a computational sustainability minor may be of little added value given typical possibilities for both computer science and sustainability-relevant minors.

While we know of sustainability majors, and sustainability-rich engineering disciplines, we know of no computational sustainability majors per se, nor do we expect these to be offered anytime soon. Nonetheless, we set ourselves the task of defining what a computational sustainability major might look like at our institution, because like so many other institutions, our students have an option of creating their own major (e.g., see Fehlen, 2016; or the readers can simply Web-search 'create your own major').

A relevant benefit of this exercise, even if just a thought experiment, is that we surveyed our institution's across-campus course offerings in search of computational relevance, and uncovered courses that clearly integrated computing into sustainability disciplines. In many cases these would be desirable electives for our computer science students, and they give good ideas to instructors for material, as noted in Section 1.

A relevant question in advising a student on a possible computational sustainability major is whether a customized major is needed, or can the desired goals be achieved within the confines of a computer science major. Our institution, and we expect all institutions, will want a reasoned answer to this from a student proposing a customized major. A related question is how close to satisfying the computer science major requirements will a computational sustainability major be. If too close, then perhaps a special major is not required, but still, the exercise of thinking through the possibilities can inform the advice we give students on selecting electives in what is otherwise a “standard” computer science major.

Short of a major, but more ambitious than a single course, is the deliberative integration of sustainability into a course sequence. We know of no implemented examples in computing per se, but Cui (2010) recommends a strategy that appears to be similar to this one relative to Green IT courses. An implemented example in engineering more generally is a sequence known as PRAXIS (1-2-3) at the University of Toronto (Heeney & Foster, 2010). This is a first-year design sequence in which sustainability is deliberatively embedded, and as a first-year sequence, it then influences the remainder of the curriculum.

5. Concluding Remarks

We have introduced a framework for considering the integration of sustainability into computing education, for reasons outlined in reports by the National Academies (NAP, 2012) and CCC/CRA (2011). This framework defines a continuum in the granularity in which integration occurs, from (a) course components, and within this, variation from individual exercises to week-long modules; to (b) individual courses in computational sustainability; and to (c) larger curricular constructs, such as course sequences and customized majors. This framework is similar to earlier proposals (e.g., Cui, 2010), but we have expanded the call for integration of sustainability to include higher order effects of computing beyond Green IT.

Moreover, we have also significantly populated the framework, particularly at the course level, and pointed to online resources that instructors can add and use content at the component level, ideally accompanied by citation practices that strengthen community. The supplemental sites are a valuable resource, if used. Our survey and highlight of the crowdsourcing mechanisms are additional new contributions of this paper.

Finally, we have made a bare start at organizing and visualizing course topic structure, which we believe can be a valuable tool for syllabus and curriculum design, particularly as more computational sustainability courses come online and can be included in analysis.

6. ACKNOWLEDGMENTS

Supported by NSF Award # 1521672 Collaborative Research: CompSustNet: Expanding the Horizons of Computational Sustainability. This technical report expands on a column that appears in IEEE Intelligent Systems (Fisher, et al, 2016).

7. REFERENCES

[1] (Abernethy & Treu, 2014) Abernethy, K., & Treu, K. (2014) “Integrating sustainability across the computer science curriculum” (2014), Journal of Computing Sciences in Colleges, Vol 30 Issue 2, pp. 220-228. Retrieved from http://dl.acm.org/citation.cfm?id=2667464.

[2] (Brunskill & Lesh, 2010) Brunskill, E., & Lesh, N. (2010), “Routing for Rural Health: Optimizing Community Health Worker Visit Schedules”, AAAI Spring Symposium 2010, https://www.aaai.org/ocs/index.php/SSS/SSS10/paper/viewFile/1139/1369.

[3] (BUCS504, 2012) “MET CS - 504 Green Information Technology,” Computer Science, Boston University, Retrieved from http://www.bu.edu/csmet/files/2012/01/CS-504-B1-Spring-2012.pdf.

[4] (CCC/CRA, 2011) “Science, Engineering, and Education of Sustainability: The Role of Information Sciences and Engineering”. Computing Community Consortium of the Computing Research Association. Retrieved from http://archive2.cra.org/ccc/files/docs/seesit/RISES_Workshop_Final_Report-5-10-2011 and http://archive2.cra.org/ccc/visioning/visioning-activities/sees-it.

[5] (Cui, 2010). Cui, Y. (2010) Integrating Sustainability into Undergraduate Computing Education, In SIGCSE-10, ACM, Retrieved from http://dl.acm.org/citation.cfm?id=1734439.

[6] (Ecowatch, ongoing) Media Watch on Climate Change, Retrieved from http://www.ecoreserach.net/climate/#4.

[7] (Fehlen, 2016) Fehlen, D. (2016). “Should I Create My Own Major?” Education Insider News Blog, Study.com, http://study.com/articles/Should_You_Create_Your_Own_Major.html.

[8] (Fisher, 2007) Fisher, D. (2007), Intelligence in Context. Presentation at NSF, March 2007, slides 7-10, Retrieved http://www.vuse.vanderbilt.edu/~dfisher/Talk.NSF.3.21.07.ppt.

[9] Fisher, D., Bian, Z., and Chen, S. (2016) “Incorporating Sustainability into Computing Education” In IEEE Intelligent Systems column on AI and Sustainability, V. 31, N. 5.

[10] (Franke, 2016) Franke, B. (2016), Mountain Paths, In Nifty Assignments, Retrieved from http://nifty.stanford.edu/2016/franke-mountain-paths/.

[11] (GMUCS795, 2016) “CS 795 - 001 Sustainable Computing,” Computer Science, George Mason University, Retrieved from https://cs.gmu.edu/syllabus/syllabi-spring16/CS795AydinH.html.

[12] (Gomes, 2009) Gomes, C., “Computational Sustainability: Computational Methods for a Sustainable Environment, Economy, and Society” (2009). Bridge Frontiers of Engineering Vol 39 No. 4 Winter 2009, National Academy of Engineering. Retrieved from https://www.nae.edu/Publications/Bridge/17281/17286.aspx.

[13] (Hamilton, 2015) Hamilton, M. (2015) “Learning and Teaching Computing Sustainability”, Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, p. 338 Retrieved from http://dl.acm.org/citation.cfm?id=2754850.

[14] (Heeney & Fosher, 2010) Heeney, A., Foster, J. (2010) “Integrating Sustainable Development into the Undergraduate Engineering Curriculum through a Mandatory First Year Engineering Design Course at the University of Toronto”, In The 6th Environmental Management for Sustainable Universities (EMSU), Retrieved from http://repository.tudelft.nl/islandora/object/uuid:99ed4c0a-a05a-4693-b7e8-a8d6ec10112c?collection=research.

[15] (IEEE Real World, ongoing) IEEE Real World Engineering Projects, Retrieved from www.realworldengineering.org.

[16] (Köhler & Erdmann, 2004) Köhler, A., & Erdmann, L. (2004) “Expected Environmental Impacts of Pervasive Computing" Human and Ecological Risk Assessment, 10: 831–852, 2004 Copyright. ISSN: 1080-7039 print / 1549-7680 online DOI: 10.1080/10807030490513856 http://www.ingentaconnect.com/content/tandf/bher/2004/00000010/00000005/art00008.

[17] (Mallet, 2016) MAchine Learning for LanguagE Toolkit, Retrieved from http://mallet.cs.umass.edu/.

[18] (Mankoff, et al, 2008) Mankoff, J., Kravets, R., Blevis, E. (2008) Some Computer Science Issues in Creating a Sustainable World, Earthzine, Reprinted with permission from Computer Magazine, 2008 IEEE. Retrieved from http://earthzine.org/2008/11/17/some-computer-science-issues-in-creating-a-sustainable-world/.

[19] (Merlot, ongoing) MERLOT (Multimedia Educational Resource for Learning and Online Teaching, Retrieved from http://www.merlot.org.

[20] (NAP, 2012) "Computing Research for Sustainability" (2012), Chapter: 3 Programmatic and Institutional Opportunities to Enhance Computer Science Research for Sustainability, Retrieved from http://www.nap.edu/read/13415/chapter/5.

[21] (NDSUCSCI428, ongoing) CSCI 428. Computational Techniques for Environmental Sustainability, North Dakota State University, Catalog description retrieved at https://bulletin.ndsu.edu/course-catalog/descriptions/csci/

[22] (Parlante, ongoing) Nifty Assignments Retrieved from http://nifty.stanford.edu.

[23] Patterson, D. A., & Hennessy, J. I., (2005) Computer Organization and Design: The Hardware/Software Interface, 3rd ed., Morgan Kaufmann.

[24] (Petersen, et al, 2007) Petersen, J. E., Shunturov, V., Janda, K., Platt, G., Weinberger, K. (2007),"Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives", International Journal of Sustainability in Higher Education, Vol. 8 Iss 1 pp. 16 - 33. http://www.emeraldinsight.com/doi/pdfplus/10.1108/14676370710717562.

[25] (PittCS690, ongoing) CS 90: Computers and Sustainability, University of Pittsburgh, Course title retrieved at http://www.engineering.pitt.edu/MCSI/Sustainability-Courses/

[26] (Scott, 2011) Scott, M. (2011) Wator World, In Nifty Assignments, Retrieved from http://nifty.stanford.edu/2011/scott-wator-world/.

[27] (SewaneeCSCI120, ongoing) CSCI 120 Introduction to Environmental Computing, University of the South, Catalog description retrieved http://math.sewanee.edu/courses/.

[28] (SISL CS, lapsed) Teaching Sustainability in Computer Science, Retrieved from http://serc.carleton.edu/sisl/sustain_in_cs.html.

[29] (SOUES475, ongoing) "ES 475 Environmental Modeling", Environmental Science, Southern Oregon University. https://inside.sou.edu/sustainable/programs.html#.

[30] (Tomlinson, 2010) Tomlinson, B. (2010) Greening through IT: information technology for environmental sustainability, MIT Press.

[31] (Tomlinson, 2011) Tomlinson, B. (2010). "Can More Efficient IT Be Worse for the Environment" Computer, Vol. 44, Issue: 1, pp. 87 - 89, IEEE Computer Society Retrieved from http://wtf.tw/text/more_efficiently_unsustainable.pdf.

[32] (UNMARTS441/541, 2016) ARTS 441/541, ECE, CS, SUST, CRP: Computational Sustainability, University of New Mexico, Retrieved from http://www.andreapolli.com/unm/computationalsustainability.html.

[33] (VUCS3265, ongoing) Dorm Energy project, CS 3265 Introduction to Database Management Systems, Vanderbilt University, Retrieved from https://my.vanderbilt.edu/cs265/dorm-energy-project-deliverable-1/ and https://my.vanderbilt.edu/cs265/files/2015/04/DormEnergy.UML_.StandardandRubric.pdf.

[34] (VUCS3892/5892, 2016) CS 3892/5892 Computing, Environment, and Energy, Vanderbilt University, Retrieved from https://my.vanderbilt.edu/csx892/.

[35] (VUEES4760, ongoing) "EES 4760 Agent- and Individual-Based Computational Modeling, Earth and Environmental Sciences", Vanderbilt University, Retrieved from http://www.vanderbilt.edu/scientific_computing/courses.php.

[36] (Wikibook CompSust courses, 2016) Artificial Intelligence for Computational Sustainability: A Lab Companion/List of Computational Sustainability Courses. (2016, June 29). Wikibooks, https://en.wikibooks.org/wiki/Artificial_Intelligence_for_Computational_Sustainability:_A_Lab_Companion/List_of_Computational_Sustainability_Courses

[37] (Wikibooks, AI Lab) "Artificial Intelligence for Computational Sustainability: A Lab Companion" Retrieved https://en.wikibooks.org/wiki/Artificial_Intelligence_for_Computational_Sustainability:_A_Lab_Companio