Study of Architectural Choice Impact on Software Sustainability

Ruzanna Chitchyan
Department of Computer Science
University of Leicester
University Road
Leicester, UK
+44 (0)116 252 3828
rc256@le.ac.uk

Ahmed H. Obeid
Department of Computer Science
University of Leicester
University Road
Leicester, UK
ahoo1@student.le.ac.uk

Helge Janicke
Software Technology Research Laboratory, Gateway House Building
De Montfort University
Leicester, UK
+44 (0)116 257 7617
heljanic@dmu.ac.uk

ABSTRACT
In this paper we explore how the choice of software architecture can affect software energy use and CO2 emissions – two specific issues related to software sustainability.

Keywords
Software architecture, software energy use, software sustainability.

1. INTRODUCTION
Software architecture is the structure of the system, which comprises software elements, their externally visible properties, and the relationships among them [1]. Today software architectural patterns (or styles) are used for solving some frequently repeated problems. For instance, the Blackboard style is used where a centrally maintained knowledge repository must be updated by a large number of users. The main requirements supported via this style are knowledge sharing, maintainability, changeability, and reusability of knowledge components. On the other hand, the client/server architecture supports distributed applications, where a number of (often geographically) distributed users request and use services provided by a server. This style is particularly well suited for centralised delivery of frequently used, repeated functions computed over a large dataset (e.g., carrying out money transfers for banking applications, sending emails, etc.).

In this paper we explore the effects that a choice of software architecture has on energy efficiency, and CO2 emissions properties of a software system. We use, as a case study, the Health Watcher (HW) [2] - a previously developed system, which had no specific consideration of sustainability.

Two architectural styles, both able to support the requirements of this system are realised for this case study. The resultant architectures are then evaluated for their impact on sustainability.

2. OVERVIEW OF HEALTH WATCHER
Health Watcher [2] is a web-based information system for public health monitoring and complaint registration developed and presently used in Brazil. The system allows citizens to report complaints, and query information on diseases, health service units, and previously made complaints. This case study was selected because it met a number of key criteria relevant to this study. Firstly, HW is a real and non-trivial system and so enables credible conclusions to be drawn. Secondly, the HW system has been developed without explicit note of sustainability, allowing for such analysis to be introduced. Finally, the original requirements are represented as use-cases, which are publicly available. Besides, HW has also been used in a variety of empirical studies [5, 6], allowing for future work on integration of current findings with past study results.

3. HEALTH WATCHER: Study of Alternative Architectures
The current implementation of the HW system uses the client-server architectural style. Yet, the client-server architecture has 2 quite distinct variants – the thick and thin client architectures. We set out to see how differences in sustainability-related requirements will result if one or the other of these two flavors of the client-server style were used. Figures 1 and 2 respectively present these styles for the HW system

a) The **thin client** version of the architecture is presented in Figure 1. Here the architecture contains 3 main components: the client who only processes a display-capability, the server that hosts the application logic, and the database that hosts the generated data. The client has no processing capability, so all processing will take place at the application server side, with data served through database server. Thin client cannot be operational when disconnected from the application and database servers.

b) The **thick client** version of the client-server architecture is presented in Figure 2. Here the application logic and GUI are deployed on the same physical machine on the client side. The client also maintains a local database. This database is synchronized with the main database located at a different physical machine. At the same time, this style retains a local database, which allows for an off-line data access, when it is necessary.

Measuring the influences of how an architectural style can affect the system sustainability is important for selecting the most sustainability-inductive architecture. Two particular measures: energy efficiency and CO2 emissions are used for this study (note, response time and cost were also calculated, but are not discussed here due to space limits):

Electricity consumption calculates the power that each type of architecture will consume for the implemented system. Since each type of architecture runs on some hardware, this metric will consider two types of end user devices (one for thin client and one for thick client) and measure the power they consume per day. We take a note of processor utilisation as well, since it substantially affects energy consumption. The energy required to operate the system server will also be considered. The result of this per-day calculation, if multiplied by 365, will estimate annual power consumption. Thus:

Energy consumption = user devices consumption + system servers consumption + cooling system servers (1)

The **CO2 emissions metric** uses the previously defined energy consumption to estimate the annual CO2 emissions produced via the given architectural solution. The amount of energy consumption is multiplied by 0.65 which represent how much CO2 emissions are produced from one KW of electricity in the UK [8]. Thus:

*CO2 emission = total energy consumption * 0.65 (2)*

In this calculations we use details on hardware power consumption provided by the producers [9] as well as details on the energy utilisation provided by study from Cornell [4].
Hardware and usage parameters for the devices are: (note, here a set of samples is used, but each system should be evaluated with its own relevant data for power consumption etc.):

- thin client Dell Wyse T10 and
- Dell desktop (GX 280 + LCD monitor 17). Wyse T10 (keyboard+ 1 ps/2 mouse+ monitor) whose energy consumption is a round 7.2 Watt [8,9]
- Number of users: 50
- Number of hours worked per day: 9;
- Server power consumption 520w max, 200 w in idle (no load) mode.
- Number of users per hour: 3, each using for 3 min.
- Time between use and standby: 3 min.

The energy use per the thick/thin client-server styles, calculated based on Equations (1) and (2) is shown in Table 1 below. The calculation accounts for processor utilization during the day.

When looking at the total energy consumption by each architectural style with use of 50 client devices (as per Table 1), we observe that the thick client-server variant uses above 3.5 times the energy of the thin one. This is also a trend that is set to grow with increase of the client devices in use. However, we should also expect that due to its strong reliance on server processing, the thin client-server variant will require and additional server resource much sooner if the number of users grows substantially.

Due to our used calculation, the CO2 emissions are directly proportional to the consumed energy, so again, the thin client version will emit 3.5 less CO2 in the given scenario.

4. CONCLUSION

In this paper we set out to study how the choice of an architectural style affects two specific properties of software system's sustainability: it's energy use and CO2 emissions. We then sketched the evaluation using two versions of the client-server architecture.

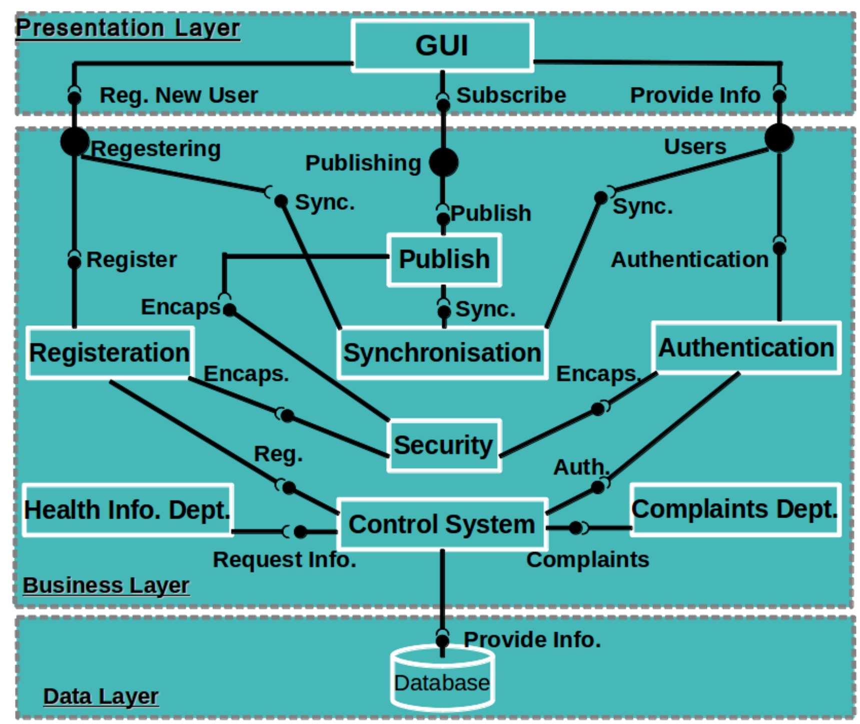

Figure 1: HW case study: thin client version of client-server architecture

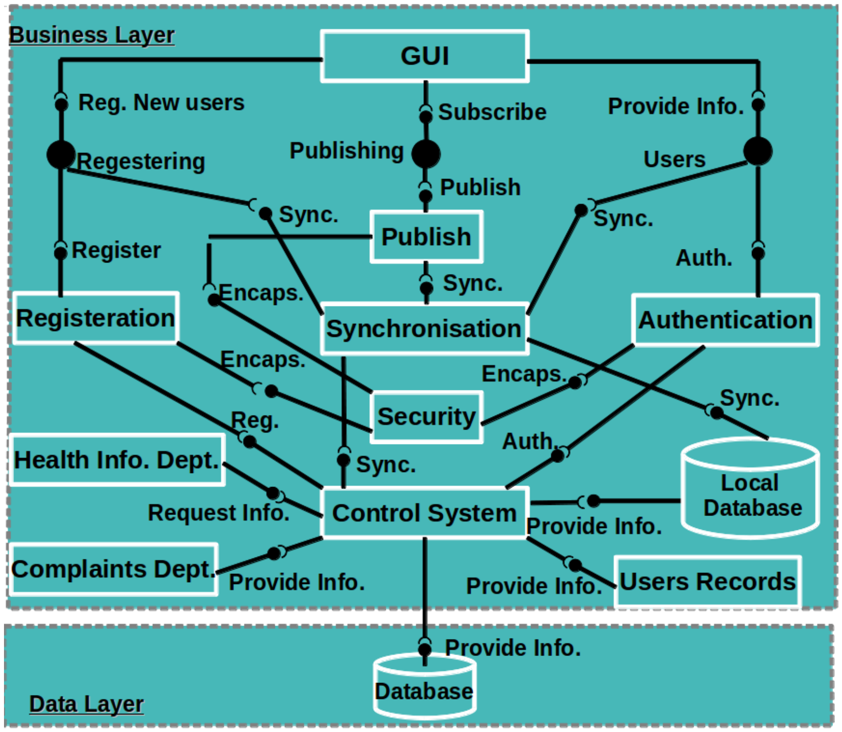

Figure 2: HW case study: thick client version of client-server architecture

Our study so far shows that there is a very clear affect that even the hardware configuration used to support a chosen style will have on the system's sustainability.

The next step in this work will be to take a finer-grain look at the architectural influence, e.g., studying it through use case-based evaluation. Moreover, additional evaluation criteria, such as response time, usability, cost should be integrated into the suite of evaluation metrics for a more representative picture for all sides (i.e., including social, economic, and environmental) of sustainability concern.

We are also aware about the need to consider a number of threats to the validity for such a study, including issues of hardware selection, number of users, etc. These concerns will be of focus in more detailed studies.

5. REFERENCES

[1] B Bass, L., Clements, P., & Kazman, R. (203). *Software architecture in practice*. Addison-Wesley Professional.

[2] S. Soares, P. Borba, E. Laureano, Distribution and persistence as aspects, Software: Practice and Experience 36 (7) (2006) 711–759.

[3] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (204). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203-236.

[4] Computer energy usage facts, university of Cornell.ULR: http://computing.fs.cornell.edu/Sustainable/fsit_facts.cfm Accessed at [12-04-2014].

[5] P. Greenwood, T. T. Bartolomei, E. Figueiredo, M. Dósea, A. F. Garcia, N. Cacho, C. Sant'Anna, S. Soares, P. Borba, U. Kulesza, and A. Rashid. On the impact of aspectual decompositions on design stability: An empirical study. In *ECOOP,* pages 176–200, 2007.

[6] Bianchini, R., & Rajamony, R. (2004). Power and energy management for server systems. IEEE Computer, 37(11), 68-74..

[7] J. Cabot et. al: Integrating sustainability in decision-making processes: A modelling strategy. ICSE Companion 2009: 207-210

[8] M. Berners-Lee, "How bad are bananas?: the carbon footprint of everything." Greystone Books Ltd., 2011.

[9] Dell Wyse T class technical report, URL: http://www.wyse.co.uk/sites/default/files/products/spec_sheets_2012-05/Dell-Wyse-T-class-IA.pdf Accessed at [2-06-2014]

Table 1: Calculation of Energy Consumption per Architectural Style

Thick Client	Idle	Active	Standby
	116 w	175 w	2 w
Assume: 9 h working day, 3 users use system per hour and each spend 3 min per query. The system then stays in idle more for 3 minutes then goes to stand-by mode.	81 minutes (3*9*3=81)	81 minutes (3*9*3=81)	21.3 hours (24h - 2*81 min)
Client Energy use per state per a day	116w *1.35h=156.6 w/h	175w*1.35h=236.25w/h	2 w * 21.3 h= 42.6 w/h
Client Energy use per a day	156.6 + 236.25 + 42.6 = 435.45 w /d		
Server with Thick Client	Average (40% utilization)	Active	Idle
Note: given consumption for the no load and max load states the average utilization at 40% (n=40%) is calculated as (Pmax–Pmin)* n/100 + Pmin [6,9]	(520–200)*40/100 + 200= 320* 0.4 +200= 328 w	520w	200w
Assume: same time in each state as the client device (above), however, *the server with a thick client does not use max utilization in active state, but uses the average rate.*	81 minutes	81 minutes	21.3 hours
Server Energy use per a day	2.7 h * 328 + 21.3h*200 = 5,145.6 w per day		
Thick Client-Server: total energy per day with 50 client devices	435.45 *50 + 5,145.6= 26918.1 w per day		
Thin Client	Idle time	Active time	Standby time
Assume: 9 h working day, 3 users use system per hour and each spend 3 min per query. The system then stays in idle more for 3 minutes then goes to stand-by mode.	81 minutes (3*9*3=81)	81 minutes (3*9*3=81)	21.3 hours (24h - 2*81 min)
Because there is no research found explained how much thin client consume power in idle and standby state we assume at idle it consume 40% of its power consumption and 20% at standby state	7.2 w * 0.4 = 2.88 w	7.2 w	7.2 w * 0.2 = 1.44 w
Energy use per state per a day	2.88 w * 1.35 h=3.88 w/h	7.2w * 1.35 h= 9.72 w/h	1.44w*21.3h= 30.672w/h
Total energy use per day	3.88 w+ 9.72 w + 30.672 w = 44.272 w/d		
Server with Thin Client	Average (40% utilization)	Active	Idle
Note: given consumption for the no load and max load states the average utilization at 40% (n=40%) is calculated as (Pmax–Pmin)* n/100 + Pmin [6,9]	520–200)*40/100 + 200= 320* 0.4 +200= 328 w	520w	200w
Assume: same time in each state as the client device (above), however, *the server with a thin client uses max utilization in active state, but uses the average rate.*	81 minutes	81 minutes	21.3 hours
Server Energy use per a day	1.35 h * 328 + 1.35*520+21.3h*200 = 5,404.8 w per day		
Thin Client-Server: total energy per day with 50 client devices	44.272 w *50 + 5404.8 = 7,618.4 w per day		