Thermal Management of Software Changes in Product Lifecycle of Consumer Electronics

Yoshio Muraoka, Kenichi Seki, Hidekazu Nishimura

Thermal Management of Software Changes in Product Lifecycle of Consumer Electronics

Yoshio Muraoka[1], Kenichi Seki[2], and Hidekazu Nishimura[1]

[1] Graduate School of System Design and Management, Keio University, Japan

[2] SDM Research Institute, Keio University, Japan

Abstract Because the power consumption of consumer electronic products varies according to processor execution, which depends on software, thermal risk may be increased by software changes, including software updates or the installation of new applications, even after hardware development has been completed. In this paper, we first introduce a typical system-level thermal simulation model, coupling the activities within modules related to software, electrical parts, and mechanical structure. Then, we investigate a case study of thermal management in both the development and maintenance phases of the product lifecycle. Reusing the simulation model, the thermal risk of software changes that may cause an enormous number of variations can be efficiently evaluated.

Keywords: Product lifecycle management, Thermal design, SysML, Software change, Electronic products, Embedded system, System level, Simulation, Collaborative design, Low-temperature burn injury

1 INTRODUCTION

1.1 Thermal design of electronic products

Because of the demand for small and fast processing speeds, thermal design has become increasingly difficult in the development of consumer electronic products such as smartphones, tablets, and video game consoles. Although technology for power reduction and heat spreading has been continually developed, product temperatures have become high during operations. For example, handling content-rich applications such as video games and shooting high definition video cause high temperature rise (Consumer Reports 2012, Qualcomm 2012), and this becomes a major issue for product quality. Temperatures exceeding the optimum operating temperature range of a component may damage the product. Moreover, high en-

closure temperatures induce low-temperature burn injuries if the user touches the product. Fig. 1 shows a diagram of an electronic product that has several layers of modules. In the modules, the semiconductors within the electrical parts dissipate heat based on the software. Because mechanical structures spread heat, the objects targeting in thermal design include all the modules shown in the figure.

Because many features of electronic products are flexibly implemented by software, product complexity increases (Siemens 2012). Product variation now occurs mostly after purchase by the installation of software, including applications and new operation system (OS) (Kuusela 2012). The number of available Android application has already reached one million, and the Android OS is updated several times a year at least (Google 2013).
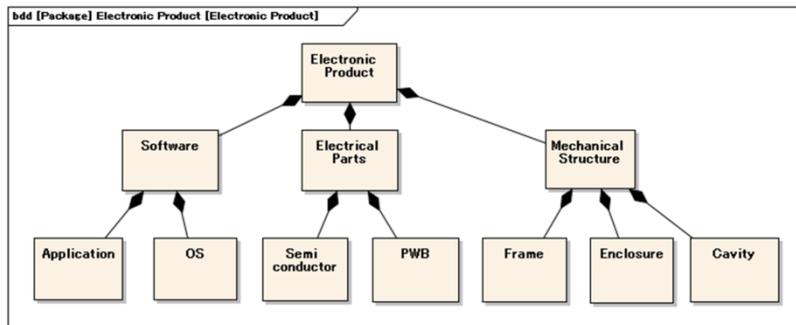

Fig. 1 Diagram of modules in an electronic product

Fig. 2 shows an example of the product lifecycle of electronic product. After product planning is authorized, the product system is designed, verifying that the planned system can meet the requirements. For example, temperatures must not exceed the target for product quality. In a design approach for systems engineering, architecture design initiates in the concept phase of the lifecycle. An outline of the system is then checked in detail, decomposing elements into physical specifications. In general, software, electrical parts and mechanical structure are developed by different engineering teams. These modules are developed in parallel; however verification is necessary to ensure that the functions of system work to satisfy the requirements. For thermal verification, thermal simulation is often used besides temperature measurement, especially when the modules are being developed. Before mass production, the designed modules are integrated into physical prototypes. Quality Assurance (QA) tests are conducted, and modules that fail these tests are improved during further development. Verification must be performed at the correct time, considering that modules vary in the level of progress during their developments. In addition, software also changes in the maintenance phase, i.e., after hardware development is complete. As cited above, product variation now occurs mostly after distribution. In this study, a simulation model is used to verify the architecture design during development. Further, the simulation model is also used for maintenance during the product lifecycle.
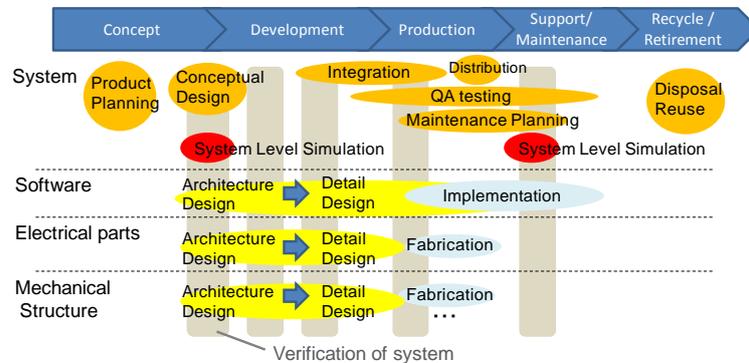
Fig. 2 Example of the product lifecycle of electronics product

1.2 System-level design approach

As a collaborative design to prevent thermal problems, this study proposes a system-level simulation for the system consisting of different engineering domains. The system-level design approach is also known as a model-driven development method. To achieve a coordinated design within multi-disciplinary design teams, Balmelli (2007) proposed model-driven systems development using systems modeling language (SysML). This approach adopts operational, functional, and physical viewpoints to separately address different engineering concerns, while maintaining an integrated representation of the underlying design.

For the collaborative design of consumer electronics, Seki et al. (2011a) investigated a module-based design approach using SysML. Their study used initial target values (ITVs) as tentative boundary conditions for assigning different design sites and refined them during product development. This approach reduced inconsistent performance of the modules that resulted from a lack of communication between design sites and the work iterations required to correct such inconsistencies when applying ITVs to independent module design. In addition, Seki et al. (2011b) proposed a thermal design approach on the basis of the design structure matrix (DSM) to systematically assign ITVs. In the study, ITVs were created between module designs of hardware, pertaining to electrical parts and mechanical structures, particularly those including cavities.

By assigning ITVs to software modules, Muraoka et al. (2013a) investigated a system-level thermal simulation model. While the temperature that causes low-temperature burn injuries is quantified, thermal risk is evaluated based on the calculated temperature. A simulation model for resolving conflicting user requirements was developed by changing the design parameters. In addition, Muraoka et al. (2013b) applied this approach to develop design specifications with regard to a software change. ITVs were used to prepare alternative scenarios as countermeasures. These approaches were aimed at developing design specifications in the de-

velopment phase. In this study, a system-level thermal model is used during verification of both the development and maintenance phases in the lifecycle. We also investigate a case study to improve the verification workflow.

2 SYSTEM-LEVEL THERMAL MODEL

2.1 Temperature rise of an electronics product

This study investigates the temperature rise of a simple electronic product. The mechanism by which the temperature rises in the product is illustrated in Fig. 3. Electrical parts such as semiconductors are affixed to a printed-writing board (PWB) that is fixed to a mechanical structure. Activities that cause the temperature to rise are also shown. Semiconductors are activated during the product's operation (a1). During the device's operation, software executes in the semiconductor and the load is processed at a specific frequency. Because, the consumed power is dissipated as heat, we consider that heat is generated by operations within the semiconductor (a2). This generated heat is then transmitted to the surface (a3).

Temperature increases in electronic products create various problems such as parts failure, system defects, and burn injuries to the user. As electronic products reduce in size, while their capabilities increase, surface temperature rises related to burn injuries become an urgent concern (Roy 2012). Moritz and Henriques (1947) investigated low-temperature burn injuries on human and pigskin. They established that the minimum temperature that caused burns over a six-hour contact period was 44 °C. Contact temperatures sufficient to cause a burn injury are plotted as a function of time in Fig. 4. During a short usage period, the temperature rise during operation probably will not cause a burn injury. However, at a temperature as low as 44 °C, users may not regard the device as uncomfortably hot and may unconsciously continue to operate or hold the heated product. In the following thermal design case, the maximum temperature limit depends on the required usage time. Here, we ignore the temperature rise time because this is considerably shorter than the required usage time.
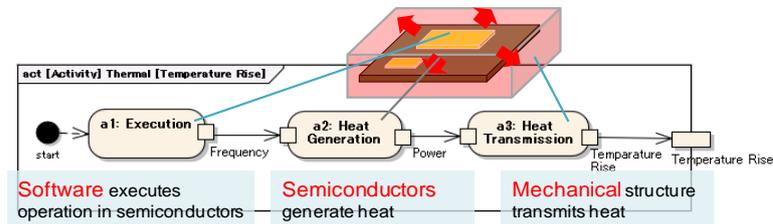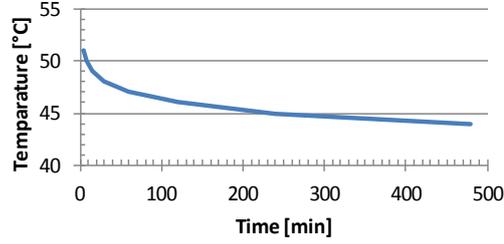

Fig. 3 Temperature rise in an electronic product

Fig. 4 Temperature that causes a low- temperature burn injury as a function of contact time (Adapted from Moritz 1947)

2.2 System-level simulation model

In this section, the temperature rise is modeled on the internal variables of the module design and ITVs, which constitute boundary conditions between modules. We describe each activity shown in Fig. 3, and its numerical calculation. The first activity is load assignment, which sets the frequency, F, of the semiconductor. The software should be designed to optimally operate at the specified processing frequency. The second activity involves heat generation. Power, P, is calculated using Equation (1) that includes internal variables, namely the physical semiconductor capacitance, C, voltage, V, and frequency, F, assigned by the software.

$$P = C \cdot F \cdot V^2 + P_{charge} + P_{static} \tag{1}$$

This equation is simplified and based on the switching power dissipation of the CMOS (Chandrakasan et al. 1992). It also assumes that semiconductors are fully activated at the specified frequency. P_{charge} is power consumption caused by voltage conversion loss during charging. P_{static} is the power consumption of the product, assumed static; i.e., it excludes the dynamic power required by the processor and charging, but includes power consumption of other components such as a power management IC and display.

The final activity concerns heat transmission. Heat transmission in a cross-sectional view of the product is shown in Fig. 5. Heat is generated at the boundary between the top and bottom parts, and it is dispatched to ambient air from both the top and bottom surfaces. This heat transmission can be described using thermal network analysis, a commonly used method for estimating the temperature rise in a simplified system (Hatakeyama 2010, Muraoka 2013).

In SysML, a parametric diagram is used to simulate and evaluate whether the constraints are satisfied. The calculation process of surface temperatures, T_{surf_1} and T_{surf_2}, using internal design variables is shown in the figure. The internal design variables are assigned to values in the modules. These values are used to calculate the ITVs transmitted between activities. In the diagram, three constraint properties are related to each module design and activity. Each constraint property is described by an algorithm that computes the related equations stated above. The

ITVs influence the module designs. Each module is designed to satisfy a target ITV using the internal design variables.
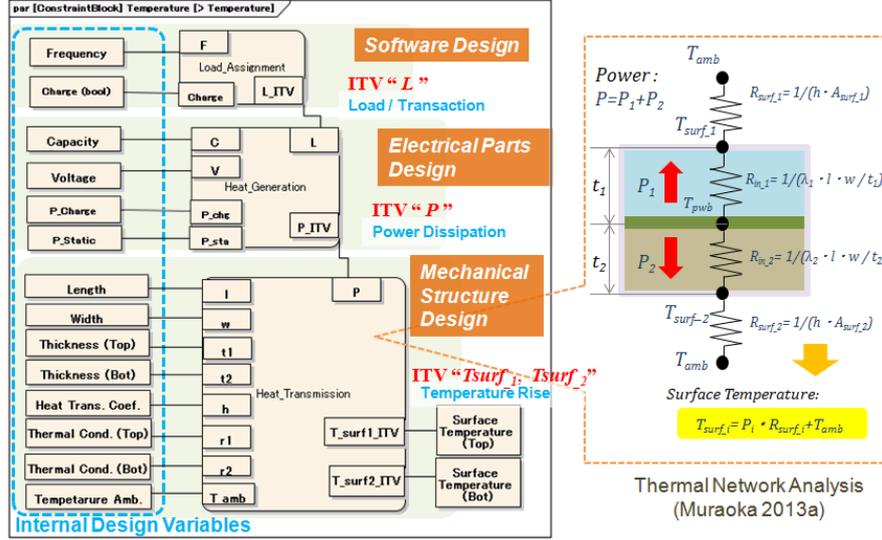

Fig. 5 Parametric model to simulate the temperature rise

3 CASE STUDY

3.1 System-level design approach in development and maintenance

In the design process shown in the DSM of Fig. 6, we introduce a system-level design approach using ITVs. In the early stage of the product development phase, system-level thermal simulation is applied to develop a product design specification (Muraoka et al, 2013b). ITVs are calculated to satisfy the requirements. Module design then starts to utilize the assigned ITVs.

Even after hardware development is complete, software changes often occur in the maintenance phase. Fig. 7 describes the verification process using a system-level thermal model. After the outline of the software change is seen in maintenance planning, thermal risk can be estimated using calculated ITVs. The effect of the software change for the ITVs can be presupposed as a change in processor frequency considering the load of task execution. Other power estimation techniques can also be used. It has been studied in different approaches such as bytecode profiling (Hao 2012) and modeling of component behavior for various applications (Zhang 2010). If the software prototype is available, power consumption can be measured using the hardware developed. Because measuring the temperature re-

quires time for preparation and data sampling during the temperature rise, one case takes a few hours, in general. Therefore, the verification time and effort can be saved by estimating the temperature using system-level simulation to filter the level of thermal risk. System-level simulation is based on simple and fast calculations. The number of measurements is limited only for critical cases. The simulation model can be the same as that used in the development phase or updated for a more accurate simulation approximating the developed system.
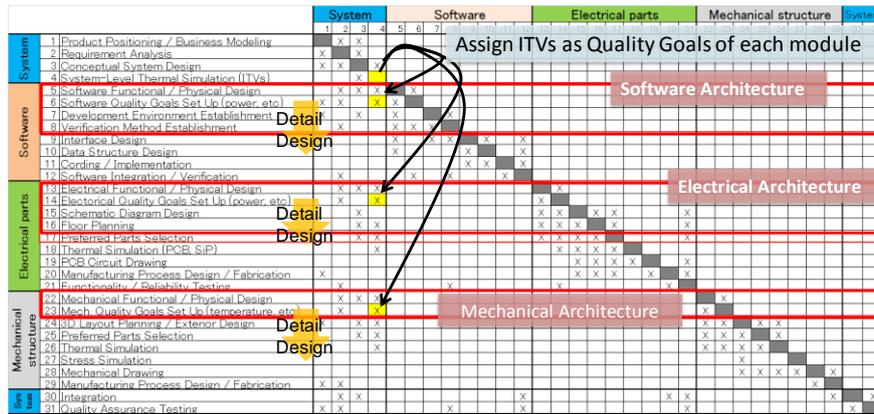

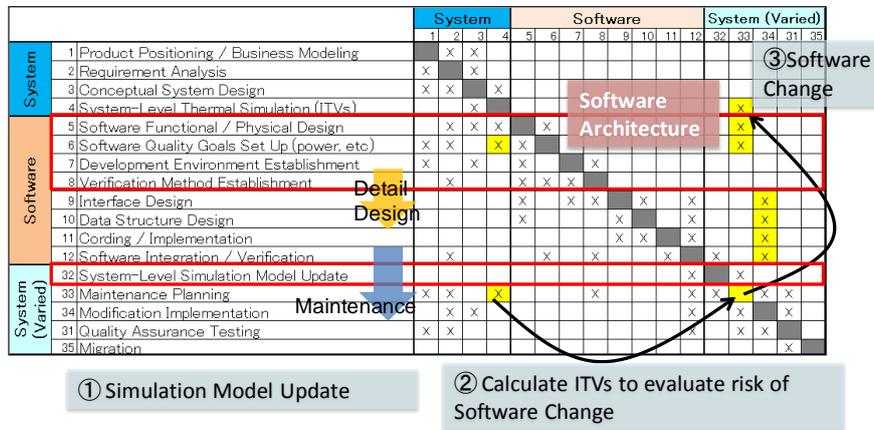**Fig. 6** Design structure matrix (DSM) in the development phase

Fig. 7 Design structure matrix (DSM) in the maintenance phase

3.2 System-level simulation

As a case study, we developed the design specification of electronic products to satisfy a target temperature. To prevent a burn injury, the surface temperature

should be below 44 °C (Moritz 1947). Table 1 provides the initial design parameters for small size electronic products. The top part of the product consists of various components such as a display and speaker, as well as an air gap for assembly tolerance. The bottom part consists of hardware such as PWB, semi-conductors, and a battery. Table 1 also provides the calculated ITVs for a certain application and battery charging. For this condition, the temperature is below the target 44 °C. Then, each module design can start to satisfy the ITVs.

Table 1. Initial design parameters and ITVs in development phase

Software	F	0.9 GHz
Electrical Parts	C	0.4 μF
	V	2 V
	P_{charge}, P_{static}	0.4, 0.9 (W)
Mechanical Structure	l, w, t_1, t_2	120, 60, 5, 5 (mm)
	h	11 W/(°C·m^2)
	λ_1, λ_2	0.1, 2 W/(°C·m)
Environment	T_{amb}	25 °C
ITVs	F	0.9 GHz
	P	2.74W
	$T_{surf\ 1}, T_{surf\ 2}$	34.6, 43.1 (°C)

During maintenance planning, thermal risk for a new software application is studied. As shown in Table 2, the new application requires a higher processor frequency to execute more tasks, and the ITV, F, rises to 1.2 GHz from 0.9 GHz. Because the power consumed by the processor is increased, the surface temperature exceeds the target 44 °C, temperature which causes a low temperature burn by touching for 98 min, as approximated by the temperature curve in Fig. 4.

To reduce the thermal risk in a new application, two alternative scenarios for countermeasures are prepared. Scenario 1 is a software improvement that executes the application at a lower processor frequency, optimizing task processing in the software. Scenario 2 is a disable function that is available to use at the same time. The use case described above is for executing an application and charging a battery. By disabling charging, the power consumption of voltage conversion loss is reduced. Proposed ITVs for each scenario are provided in table 2. In addition, a workflow for this verification is introduced in Fig. 8. If the calculated temperature exceeds the target, then thermal risk is evaluated, considering factors such as the probability and duration of usage. If the general usage time of the new application is longer than 98 min, the temperature rise should be mitigated even when there is no available hardware solution after distribution. Otherwise, the implementation of the application should be stopped at the point of QA. However, although the two countermeasure scenarios in Table 2 reduce temperatures, the temperature in scenario 1 still exceeds the target. If it is very rare to use this application for more than 204 min, thermal risk can be considered to be much reduced by this counter-

measure. Scenario 1 requires a software improvement effort, and still has some risk for prolonged usage. Scenario 2 is inconvenient for users. It may result in running out of battery power, which disables all the features of the product. Using system-level simulation, a number of scenarios can be proposed and the most practical solution can be chosen at different points.

Table 2. ITVs developed in the maintenance phase

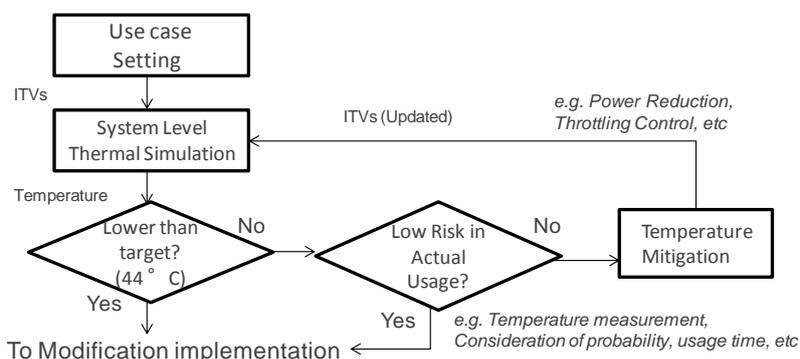|		New App	Scenario 1	Scenario 2
Countermeasure		-	Execute at a lower frequency	Disable charging
Simu-lation result	ITV F	1200 MHz	1100 MHz	1200 MHz
	P	3.22 W	3.06 W	2.82 W
	$T_{surf\ 1}, T_{surf\ 2}$	36.3, 46.3 (°C)	35.7, 45.2 (°C)	34.8, 43.6 (°C)
	Allowable us-age time	98 min	204 min	>480 min

Fig. 8 Verification workflow

4 CONCLUSIONS

This study introduces a verification approach for electronic products in both the development and maintenance phases of the product lifecycle. In system-level simulations, ITVs are calculated as boundary conditions within module designs, which comprise software, electrical parts, and mechanical structures. We investigate the use of a thermal simulation model based on SysML. The effect of software changes is evaluated as thermal risk with a quantified temperature that causes low-temperature burn injuries. As a case study, we also investigate a verification workflow. Using the simulation model, countermeasures are proposed in the event that thermal risk is increased. By estimating the risk in advance, a number of measurements can be limited. This system-level approach makes effi-

cient verification beneficial, especially for large numbers of product variations that are caused by software changes in the maintenance phase.

References

Balmelli L (2007) An Overview of the Systems Modeling Language for Products and Systems Development. J Object Technology 6(6):149–177

Chandrakasan A, Sheng A, Brodersen W (1992) Low-power CMOS digital design. IEEE J Solid-State Circuits 27(4):473–484

Consumer Reports (2012) Our test finds new iPad hits 116 degrees while running games. http://news.consumerreports.org/electronics/2012/03/our-test-finds-new-ipad-hits-116-degrees-while-running-games.html. Accessed 17 Feb 2014

Qualcomm (2012) Snapdragon S4 Thermal Comparison and Butter Benchmark. http://www.qualcomm.com/media/videos/snapdragon-s4-thermal-comparison-and-butter-benchmark. Accessed 17 Feb 2014

Siemens (2012) Role of PLM in the Software Lifecycle. White Paper. http://m.plm.automation.siemens.com/en_us/Images/15825_tcm1224-75783.pdf. Accessed 17 Feb 2014

Kuusela J (2012) How variation changes when an embedded product ceases to be embedded?. In: WICSA/ECSA 2012, Companion Volume, P 147

Google (2013) In: Google Press Event, San Francisco, 24 July 2013

Hatakeyama T, Ishizuka M, Nakagawa S (2010) Estimation of maximum temperature in 3D-integrated package by thermal network method. In: Electronics Packaging Technology Conference (EPTC), p68

Hao S, Ding L, Halfond WGJ, Govindan R (2012) Estimating android applications' CPU energy usage via bytecode profiling. In: Green and Sustainable Software (GREENS), doi: 10.1109/GREENS.2012.6224263

Moritz A, Henriques F (1947) Studies of Thermal Injury: II. The Relative Importance of Time and Surface Temperature in the Causation of Cutaneous Burns. Am J Pathology 23: 695–720

Muraoka Y, Seki K, Nishimura H (2013a) A system level thermal design approach of electronics products to resolve conflicting user requirements. In: Asia-Pacific Council on Systems Engineering 2013 (ACPOSEC 2013), Yokohama, Sep 2013

Muraoka Y, Seki K, Nishimura H (2013b) A System-Level Thermal Design Specification Development of Electronic Products Considering Software Changes. In: International Conference on Engineering Design 2013 (ICED13), Seoul, Aug 2013

Roy S (2011) An Equation for Estimating the Maximum Allowable Surface Temperatures of Electronic Equipment. In: Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 2011 27th Annual IEEE, p54

Seki K, Nishimura H, Zhu S, Balmelli L (2011a) A Parametric Design Framework to Support Structural and Functional Modeling of Complex Consumer Electronics Products. In: International Conference on Engineering Design 2011(ICED11), p282

Seki K, Nishimura H (2011b) A module-based thermal design approach for distributed product development. Research in Engineering Design 22(4): 279–295

Zhang L, Tiwana B, Qian Z et al (2010) Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: CODES/ISSS '10 Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, p105, doi: 10.1145/1878961.1878982