# A pattern language for ET robot contest

MASASHI KADOYA†, TOSHIYUKI NAKANO†, TAKANORI OZAWA†, MASAHIKO WADA†,
HIROKI ITOH†, HIRONORI WASHIZAKI†, YOSIAKI FUKAZAWA†

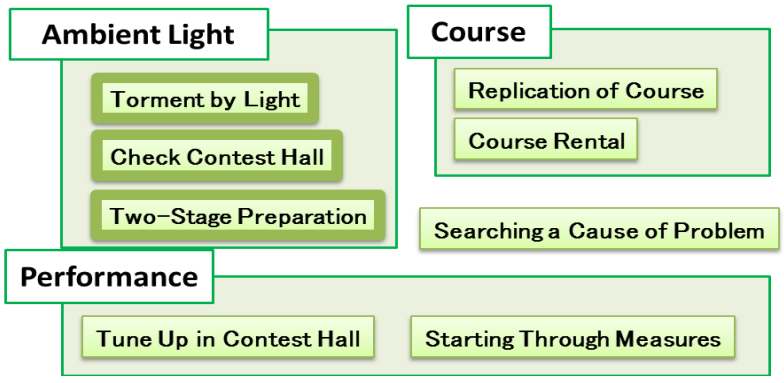
The ET robot contest is software design competitions. Because ET robot contest provide young engineers or students an opportunity to learn embedded software engineering, many people are first time participants and do not know how to develop. Consequently, we have extracted a pattern language consisting of 40 patterns from our technical knowledge and experience on the contests. We name this pattern language ***ET Robocon Strategy***. This pattern language, which is abstractly made, will coordinate with the embedded software engineering and develop a competitive contest for participants and young engineers.

## 1. Pattern languages

A member of the Washizaki lab, who has participated in this type of contest, were extracted the pattern languages. From October 2011 to June 2012, we extracted a pattern language (an activity to extract the pattern languages) via workshops and interviews and named this pattern language ***ET Robocon Strategy***.

Each pattern is composed of a name, thumbnail, context, problem, force, and solution. If a further explanation is necessary, we added related patterns and case.

We determined frequent problems during development. These problems can be classified greatly into four divisions: Environment (Green color box pattern), Team (Red color box pattern), Model (Blue color box pattern), and Programming (Purple color box pattern). Environment includes failures of robots and the course setup. Team refers to issues with team members or the whole team. Model is collective know-how to promote improved ET robot contests, while Programming refers to methodology to navigate the robot through the course. Herein ***ET Robocon Strategy*** to resolve these problems, which are denoted by thick-bordered boxes, is presented. This pattern map is separated by a big box because people who use this pattern language can understand when they avail themselves of which pattern. These scene of this pattern map are composed of ‘Early Development’, ‘Milestone’, ‘Develop Process’, ‘Course’, ‘Ambient Light’, ‘Performance’, ’Bump’, ‘State’, ‘Light Sensor’, ‘A Way of Drive’ and ‘Model’. Figure 3 shows the map arranged; Environment, Team, Model and Programming.




† Waseda University

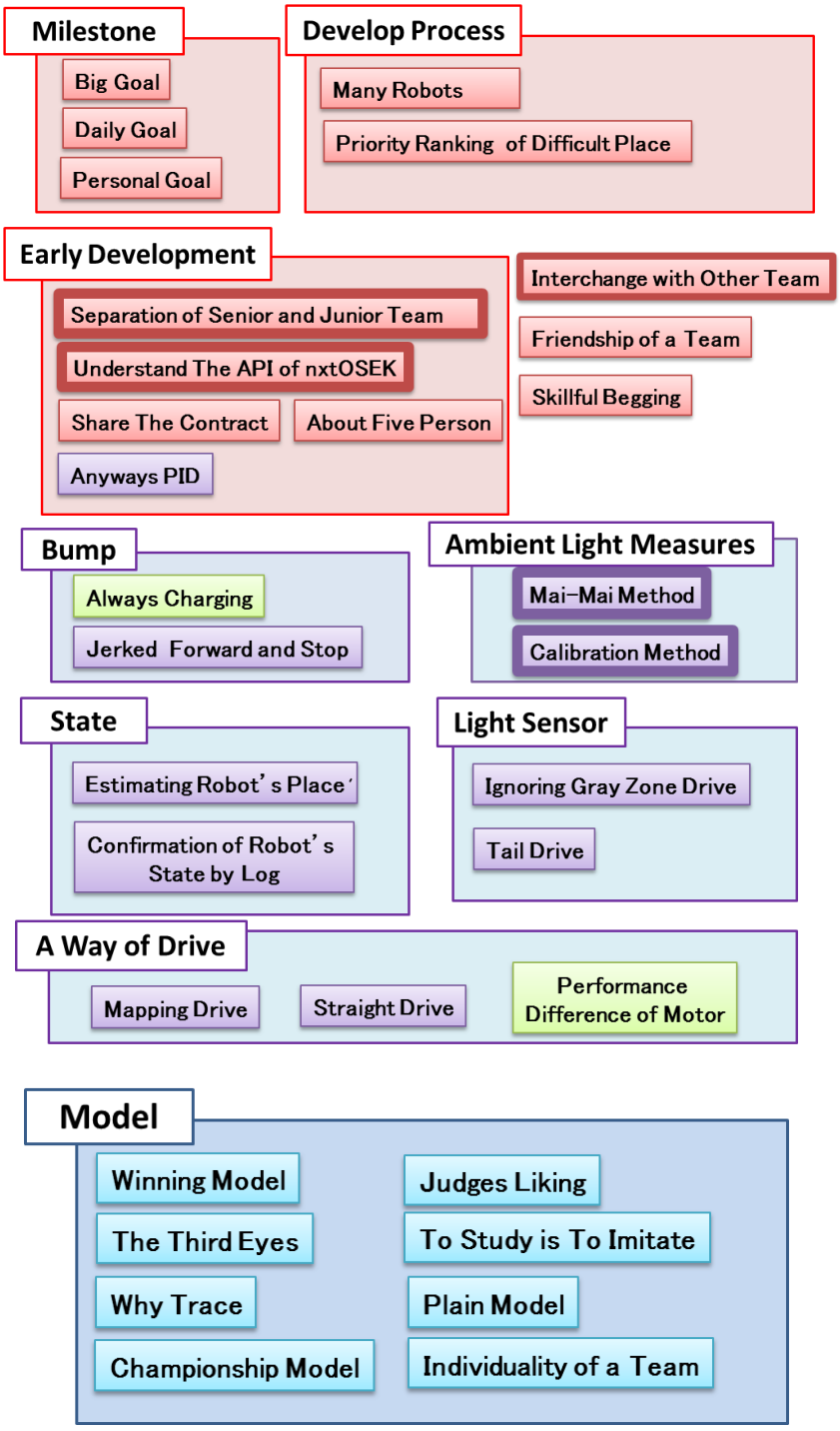
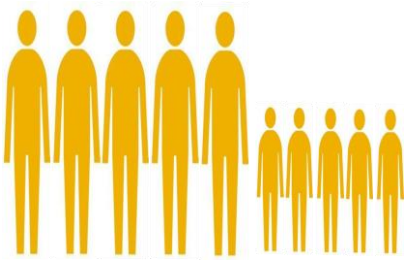



Figure 3 the map of ET Robocon Strategy

1.1. Separation of Senior and Junior Team

Form a team to avoid mixing experienced and novice members.



***Context***

A team is needed to build a robot and is constructed of many members.

***Problem***

When a team has a mixed experience levels, novice members tend to follow the senior members, making input from all team members difficult.

***Forces***

- Many people who can be different background and have a different ability could participate in this contest.
- Dividing tasks amongst the team is difficult because some tasks may be beyond the experience level of some members.
- Participants must be divided into teams and a team should consist of about five members because many member cause confusion

***Solution***

Form a team to avoid mixing experienced and novice members. Novice members actively participate, their skills and efficiency improve quickly, which results in a competitive team because an age among novice member is near (because skill is proportional to age) and novice member could tend to desire to win to the senior team.

This pattern is intended for education of novice member. Not to mention experienced team is easy to go next stage, by novice team follow experienced team, this pattern can have original value.

***Known Uses***

It was possible for a team with novice members to advance to the championship in ET robot contest 2011 because this pattern allows to a team to easily communicate and learn themselves the embedded software engineering.

***Applicability***

This pattern could be applicable to team design of every software design contests.

1.2. Understand the API of nxtOSEK

It is important to understand API of hardware before starting development for a first participant.

***Context***

It is unclear for novice member what to begin when a team wants to design the model.

***Problem***

Due to an insufficient understanding of the API, some teams are unsure where to begin designing the model.

***Forces***

- There is an opportunity to learn the API for robot because there is the website [5] that comments on hardware and ET robot contest.
- The sample code which can let robot move simply provided by an operator of ET robot contest is understandable in a tentative operational session.

***Solution***

It is important to understand the API before starting development. If the type of input is understood, then the image will be concrete. Thus, it is important for team members to review existing and sample codes.

***Applicability***

This pattern could be applicable to embedded development in the initial stage.

1.3. Exchanges with Other Teams

Through open dialogues with members of other teams, information can be discussed between different teams.

***Context***

Teams without information about the contest or access to past models are at a disadvantage. However, there are opportunities to get information in a tentative operational session and in a technical educational session.

***Problem***

How does a team acquire information and resources necessary to be competitive in the contest?

***Forces***

- Information exchange with others via the Internet should be sufficient.
- The team has as many resources or information as other team which the team wants to exchange with.
- Teams may hold information useful to other teams BUT they are in a competitive situation.

***Solution***

The team talks positively to other team in a tentative operational session or a social gathering and get information or resources. However, this pattern will not improve only by getting information or resources. The team should not only get information, but also to give members of other teams it. For example, if the team doesn't have past models and other team has it, but doesn't have the course, they get it by providing other team with their space of course. In summary, this pattern aims to grow up each other by exchanging own resource or information.

***Known Uses***

In the 2011 ET robot contest, our team has an opportunity to exchange ideas with another team. Unfortunately, a vast difference in knowledge about the contest prevented a deep exchange.

***Applicability***

This pattern should be applicable to development in the contest. In addition to previous contest winners being a valuable resource, discussions with other teams in the contest may help overcome obstacles.

1.4. Mai-Mai Method

Allow the robot to detect only the reflected ambient light.

***Context***

The team knows that there is the strong ambient light in the contest site and must complete The *Basic Stage*.

***Problem***

Many teams using the ***Calibration Method*** are unable to complete the course due to the strong ambient light in the contest site.

***Forces***

- The speed of the robot which is introduced ***Mai-Mai Method*** is slow overwhelmingly against ***Calibration Method***.

***Solution***

Measuring the value of reflected light onto the LED light sensor allows the robot to detect only the reflected ambient light. Calculating the difference between the LED ON and LED OFF states resolves this issue.

***Known Uses***

In a 2011 Tokyo meeting, a team adopting the ***Calibration Method*** was unable to complete the *Basic Stage*. In contrast, a team adopting the ***Mai-Mai Method*** safely navigated the *Basic Stage*, albeit very slowly. This pattern is extracted from a method presented by the Mai-Mai team.

1.5. Calibration Method

Allow the value to be overridden, and consequently, the robot can run earlier than using other methods.

***Context***

Issues with ambient light have been resolved. Adopting the ***Mai-Mai Method*** is not needed. The team wants faster speed.

***Problem***

A team which adopts ***Mai-Mai Method*** to the robot is guaranteed against course out, but speed of the robot is not guaranteed.

***Forces***

- The speed of the robot has a limit.

***Solution***

The robot can recognize the deflection of a measuring instrument many times in very short time via the difference in the color value by program set while running. A light sensor is important for these mechanisms because light can produce an incorrect color value. However, the ***Calibration Method*** allows the value to be overridden, and consequently, the robot can run faster than using other methods because this method let the robot move along the shortest route of the course.

***Related pattern***

This pattern is compatible with the ***Mai-Mai Method***. ***Mai-Mai Method*** is a good solution to secure stability while a robot running.

***Known Uses***

Employing the ***Calibration Method*** realizes a robot faster than the ***Mai-Mai Method***.

1.6. Check Contest Site

Visit the contest site and research the lighting conditions prior to the contest.

***Context***

The intensity of light is important for a robot because it decides a direction by light sensor. However, the team can't understand the intensity of light in the contest site which they will visit.

***Problem***

It is difficult to earnestly determine whether ambient light needs to be countered and which method (e.g., **Mai-Mai Method** or the **Calibration Method** which are the pattern to deal with ambient light**)** should be employed.

***Forces***

- Either the **Mai-Mai Method** or **Calibration Method** can be used to counter ambient light.

.

***Solution***

Visit the contest site and research the lighting conditions prior to the contest. If the lighting condition in the contest site is bad (for example, there are windows), select **Mai-Mai Method** which almost deal with ambient light. If it is not so, select **Calibration Method** which let a robot enable run faster than **Mai-Mai Method**.

***Related pattern***

If a spotlight which have a great influence on the degree of light is used in the contest site, then a ***Two-Stage Preparation***$_{(1)}$ should be adopted.

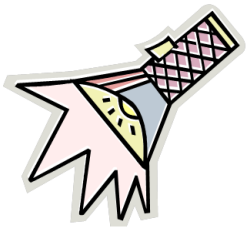
***Applicability***

This pattern can be applied to contest development and embedded development because verifying the environment should not negatively impact a robot's performance.

If visiting prior to the contest is difficult, information about the lighting conditions can be acquired by **Exchanges with Other Teams**.

1.7. Torment by Light

In an ET robotic contest, light can change the orbit of the robot.



***Context***

Even if the robot is introduced **Mai-Mai Method**, whether the robot can navigate a course with bright and dark places

must be verified.

***Problem***

The ambient light in the contest site significantly impacts the robot, which may lead to unexpected accidents.

***Forces***

- Although it costs to introduce **Mai-Mai Method**, this pattern needs more time to tune the robot.

***Solution***

Ambient light can be artificially simulated using a projector and flashlight to test the robot in the contest site, and adjustments can be made accordingly prior to the contest. If the robot gets out of the course, tune a value of the coefficient of **Mai-Mai Method** little by little in various situations. However, it is unclear whether the conditions tested for the ambient light and those of the contest site are the same.

***Related pattern***

Applying ***Check Contest Site*** can confirm the intensity of ambient light in the contest site and determine whether modifications to the robot are necessary. If ***Check Contest Site*** cannot be applied, then ***Two-Stage Preparation*** should be used.

## 2. ET robot contest

The Embedded Technology robot contest (ET robot contest) [4] is software design competitions for teams. The ET robot contest has two divisions: a racing division and a model division. In the racing division, participants design a two-wheeled robot to complete a predetermined course as fast as possible. A robot traces a black line in the course by its light sensor. Therefore, it is important that robot trace the shortest line. In the model division, documentation and software design (specification document, class diagram, development process, etc.) are evaluated. [3]

One feature of the contest is that the participants cannot touch their robot during the race. The course has two parts: the *Basic Stage* and the *Bonus Stage*. The *Basic Stage* is for speed, while the *Bonus Stage* requires that the robot successfully navigate obstacles such as a seesaw, stairs, etc. Figures 1 and 2 depict the 2012 course and robot, respectively. Another feature of the contest is that there is a tentative operational session on another day. Participants can try to run a robot several times and adjust it.

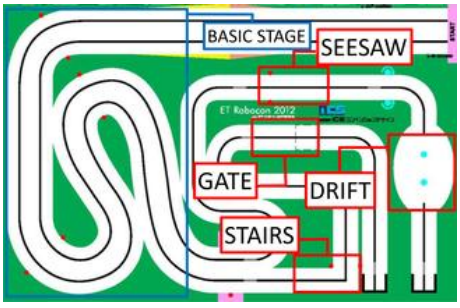



Figure 1 The 2012 course

(1) Two-Stage Preparation is a pattern that recommends both patterns from the Mai-Mai Method and Calibration Method be adopted.

Figure 2 Robot used in the 2012 contest

Below we present a potential pattern language applicable to embedded software development in these types of contests for participants and young engineers.

## 3. Conclusion and Future Work

We have extracted a pattern language consisting of 40 patterns from our technical knowledge and experience on the contests. We applied these patterns (Senior and Junior, Confirmation of API, Check Contest Site and Calibration Method) to the 2012 ET robot contest. We fielded two teams, an experienced (senior) and novice (junior) team. Although the junior team did not know where to begin, their development skills greatly improved due to advice from the senior team. Thus, the junior team applied these pattern languages to an ET robotic contest. When the junior team encountered difficulties, by discovering the appropriate Context, they were able to resolve their problems. The junior team received recognition during the contest, but did not advance to the finals. However using these pattern languages, the senior team did advance to the finals.

This pattern language for ET robotic contest is still being developed. Currently, we are collecting input using Wiki and discussing pattern languages in workshops about the contest. In the future, we plan to extract pattern languages, which can be applied to embedded development and to develop ET robot contests.

## Acknowledgement

We thank for shepherd Alan O'Callaghan for very careful review that improve this paper.

## Reference

1) Gerard Meszaros, Jim Doble, “A Patten Language for Pattern Writing”, PLoP’97.
2) Kouichiro Eto, “Origin and Evolution of Wiki”, SIGHCI 2007.
3) Hironori Washizaki, Yasuhide Kobayashi, Hiroyuki Watanabe, Eiji Nakajima, Yuji Hagiwara, Kenji Hiranabe and Kazuya Fukuda, "Experiments on Quality Evaluation of Embedded Software in Japan Robot Software Design Contest," Proc. 28th International Conference on Software Engineering (ICSE 2006), May 2006.
4) “ET robot contest 2012”, http://www.etrobo.jp/2012/gaiyou/intro.php
5) “nxtOSEK”, http://lejos-osek.sourceforge.net/index.htm