

# A New Controller for Chassis Dynamometers – Examples of Increased Usability

🔗 Chassis dynamometers, Control and instrumentation, Usability

Yasumasa Narumi

## Abstract

For computer systems, “ease of use” is often an important factor. The “ease of use” or “usability” is changed by situations in the operating site. The user needs to meet the changing requirements. Considering such factors, we newly developed a controller for chassis dynamometers.

This system has the following features:

- (1) By separating functions and User Interface (UI), the system could be flexible with the UI changes.
- (2) Adopting “rich Graphical User Interface (GUI) expressions” provides an intuitive operation.
- (3) Remote connections are made available and we provide various applications meeting operating site conditions and purposes.

## 1. Preface

Recently the computer has increasingly been important for dynamometer systems. The key requirements for computer systems are high functions, high performance, compactness, and ease of use.

To meet the above needs, we recently developed a new controller for chassis dynamometers. On its high functions, high performance, and compactness, we introduced in the Meiden Review Series No. 157, the “New Controller MDC II for Dynamometer Systems.”

This paper introduces the adopted technologies for better usability and features of this system.

## 2. Necessary Factors for Better Usability

One key factor for an easy-to-use system is an intuitive User Interface (UI). For this purpose, it is important that the operating method is common and general. In the past, a common UI was a command-line interface. It was changed to a mouse based graphical UI elements. Recently, a touch screen interface became commonly accepted. In this way, common operation methods of a user interface change with the times. Since they change according to the operating location and purposes, we need a system that can meet each particular situation.

## 3. The System Features

### 3.1 Separation of Functions and the UI

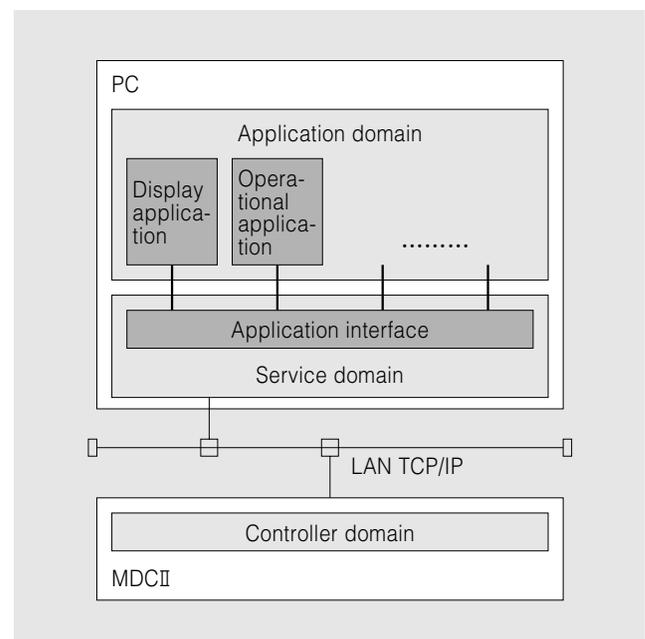
In order to offer a most suitable system for each condition, the system should be flexible in the UI changes. For this purpose, it is essential that the system be free from the impacts of UI changes to its functions and performance.

Fig. 1 shows a software configuration of this system.

It is divided into the application domain, the service domain, and the controller domain. The application domain is related with the UI, while the service and controller are related to functions and performance.

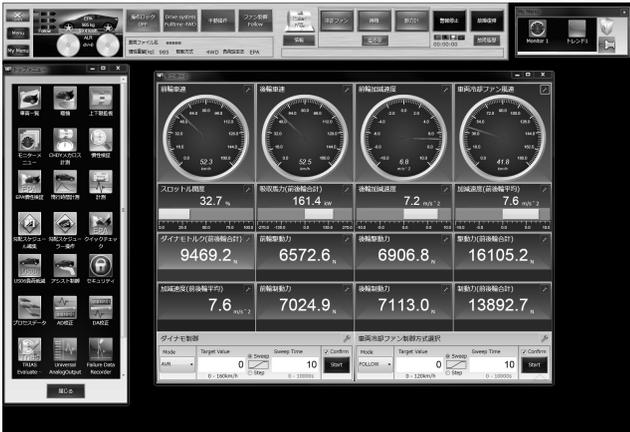
Since the service domain and its related circuits are provided with functions, these functions can be executed regardless of whether the application domain wrote its commands on the command-line interface or the Graphical User Interface (GUI).

Thanks to such a configuration, the system can flexibly cope with the UI changes.



**Fig. 1 Software Configuration**

The software configuration consists of the mutually independent application domain, the service domain, and the controller domain. This configuration allows the flexibility of GUI changes.



**Fig. 2 GUI of this System**  
Rich GUI Expressions are adopted in the UI.

**3.2 Adoption of Rich GUI Expressions**

Recently, it has been possible to make a design-conscious GUIs using such graphic expressions like animation and color gradation (“Rich GUI Expression”). At the same time, users have opportunities to see more well-designed GUIs.

Fig. 2 shows the GUI used for this system. This system adopts Rich GUI Expressions.

In order to use Rich GUI Expression, this system has adopted the Window Presentation Foundation (WPF). The WPF is a technology that produces the GUI using the GUI generation tool offered by Microsoft. The display domain is separated from the processor domain so that mutual impacts can be minimized. Because of this separation, the WPF can reinforce cooperation between a designer and a programmer.

When a designer produces display domain software using the WPF, a programmer can program a processor domain based on the deliverables from the display domain. By using the display domain produced by the designer as it is, we can realize the design intended by the designer. The programmer is not required to think how the designer’s design should be realized through the programming. This system allows each profession to reflect each ability of their expert field.

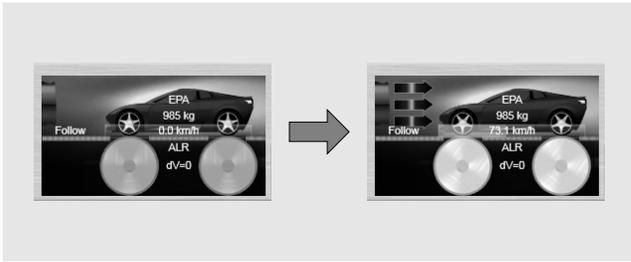
As mentioned above, we realized well-designed GUIs through the use of the WPF and cooperation between the designer and the programmer.

**3.2.1 Application Examples of Rich GUI Expressions**

For intuitive operation, we would like to introduce an example below where Rich GUI Expressions like animation etc. are used.

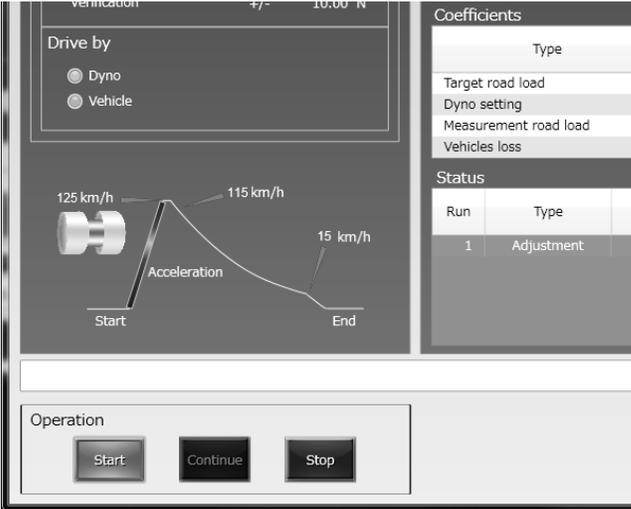
Figs. 3 and 4 show some examples of Rich GUI Expressions that enable the operator to perform the intuitive grasping of the current situation.

In Fig. 3, the example shows the status of a vehicle cooling fan and rollers. When the vehicle cooling fan is turned “ON,” an arrow mark appears. It shows the fan is operating. When the vehicle speed is



**Fig. 3 Display of Current Status at the Vehicle Cooling Fan and the Rollers**

When the vehicle cooling fan is turned “ON,” an arrow mark appears. When the vehicle speed is increased, rollers in the picture begin to rotate; this is to enable the operator to intuitively grasp the current status.



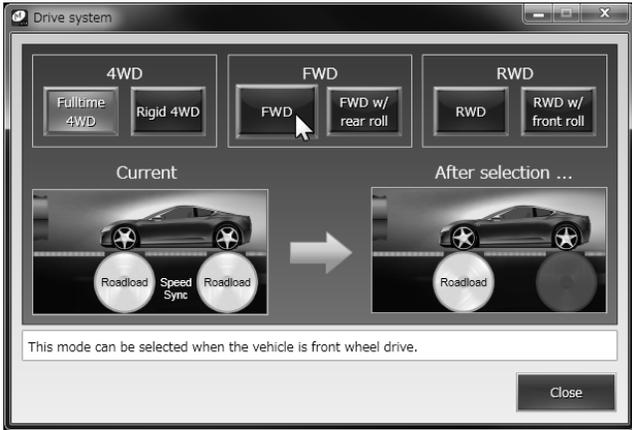
**Fig. 4 Display of Overall Work Flow and Current Update**

Overall work flow and current update can be recognized intuitively. In the case of the example in the diagram, acceleration is being carried out and the accelerating section is active with animation.

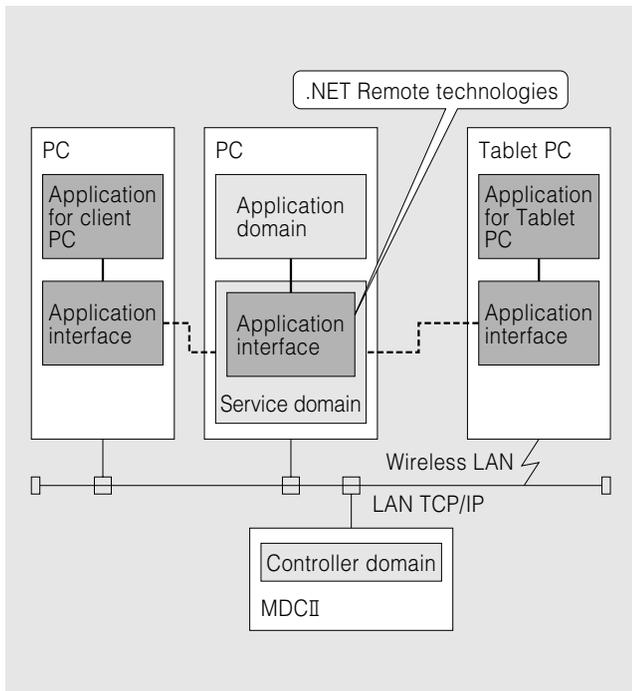
increased, rollers in the picture will rotate and it shows the rollers moving.

Fig. 4 shows the status of road load derivation. The road load derivation function is used to adjust the measured data to the target values by performing a free-wheeling operation. The road load derivation function shows the status during acceleration, measurement, and deceleration. An all process flow is shown in the diagram. It shows the current process shown in animation. When a car is accelerating in Fig. 4, we display an animation by moving a part changing color in the domain that the graph climbs into the top.

One more example is shown below. Fig. 5 shows an example of intuitive grasp of what follows after the operation. Fig. 5 shows a drive system select window. This window makes a drive system select (4WD, FWD, RWD) of the test vehicle. “Current” shows in the picture the drive system currently selected. When a mouse cursor hovers and clicks on the drive system button, the next drive system in sequence is shown in “After Selection” in the picture. And if it is clicked, the drive system should be changed. In the example diagram, as 4WD is selected, “Current” shows the front and



**Fig. 5 Display of the Post-Click Operation Mode**  
 When a cursor (mouse pointer) is put on the button icon, “what happens” is shown on the right side in the picture indicating the switching drive mode, if the mouse is actually clicked.



**Fig. 6 Example Configuration of Remote Connections**  
 Since the .NET Remoting technologies are adopted, one can use the functions offered by the service domain through the network.

rear rollers highlighted. Since the mouse is now on the FWD button, the front roller is highlighted and the rear roller is lowlighted in “After selection.”

As such, Rich GUI Expressions are used. It indicates what follows once the mouse button is clicked.

**3.3 Without Specified Operating Place**

Since this system adopts “.NET Remoting technologies,” remote operation is possible.

Fig. 6 shows an example of configuration of remote connections. Connection with the service domain is possible via network. Irrespective of wired or wireless mode, functions to be offered by the service domain can be utilized as long as any access for LAN connections is available. In this system, the location of the application domain is not an issue.



**Fig. 7 Application of Tablet PC, Example 1 (System Application)**

The same operation as in the wire-connected system can be carried out through wireless connections.



**Fig. 8 Application of Tablet PC, Example 2 (Application for Current Value Display)**

The screen shows a specialized display of current values in regard to throttle aperture, vehicle speed, and pitch for intuitive recognition.

**3.3.1 An Application Example of Remote Connections**

Recently, handheld computers with touch screen operation (Tablet PC) are widely accepted. Figs. 7, 8, 9 show some examples of Tablet PC applications through remote connections.

Fig. 7 shows an example of the same applications working on a Tablet PC as those for the system. As suggested by this example, the same operation as that for the system can be carried out through wireless connections.

Fig. 8 shows an application example at local site in showing the current value on a Tablet PC. The screen shows a specialized display of current values: throttle aperture, vehicle speed, and pitch. Tires begin to rotate as the vehicle speed is increased. According to the value of pitch, the gradient of the drawn vehicle changes in a circle at bottom right.



**Fig. 9 Application of Tablet PC, Example 3  
(Application for road load derivation)**

In the event where a driver uses a Tablet PC inside the vehicle under test, the START button icon is designed for easy pushing, and one could readily know the current status. Current conditions can be indicated in audio mode even without seeing the screen.

Fig. 9 shows an example of road load derivation application on a Tablet PC. The screen shows a specialized road load derivation function. In a situation where a driver uses a Tablet PC inside the vehicle under test, the START button icon is designed for easy pushing, and one could readily know the current status. Status under acceleration or measurement could be

announced by the voice. Its innovative design allows a driver to know the status without seeing the Tablet PC.

As described above, remote connections are made possible so that the applications can be used according to any situation at the project site, without specifying equipment, place, or purpose.

#### 4. Postscript

Easy-to-use UI can be changed according to the situation. We must offer the best UI fitting for the given situation. The following points are our UI development design policies:

- (1) Separate what needs to be changed and what should be left unchanged.
- (2) Select the UI Expression which fits the requirements of the given times, location, and purpose
- (3) Cooperation between designers and programmers

In this time, development results in a case where we could improve product usability.

Building up and drawing on this time result, we would like to work on further improvements and offer the system meeting the operating conditions in a timely fashion.

- All product and company names mentioned in this paper are the trademarks and/or service marks of their respective owners.