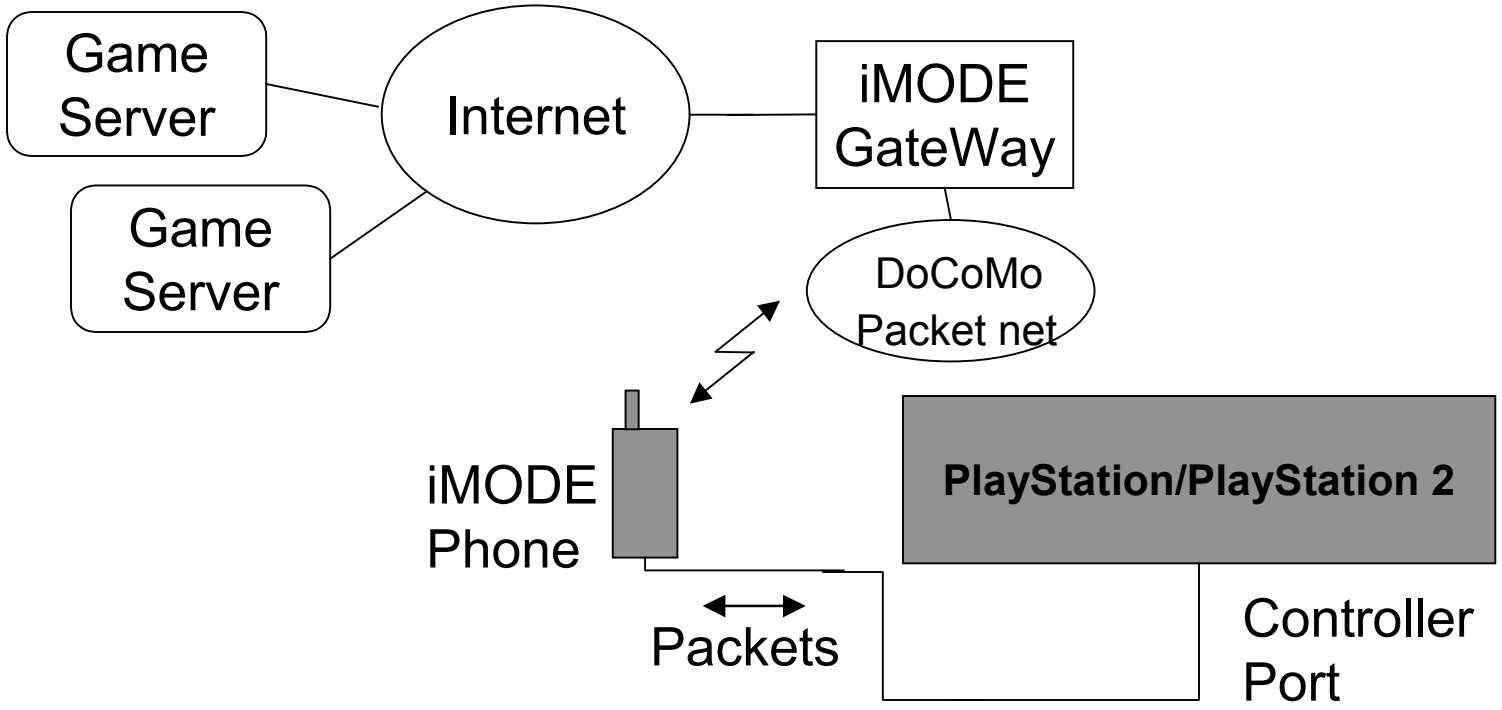
Software Development Seminar

Sony Computer Entertainment Inc.

iMODE System

Concept 1: Net Gateway

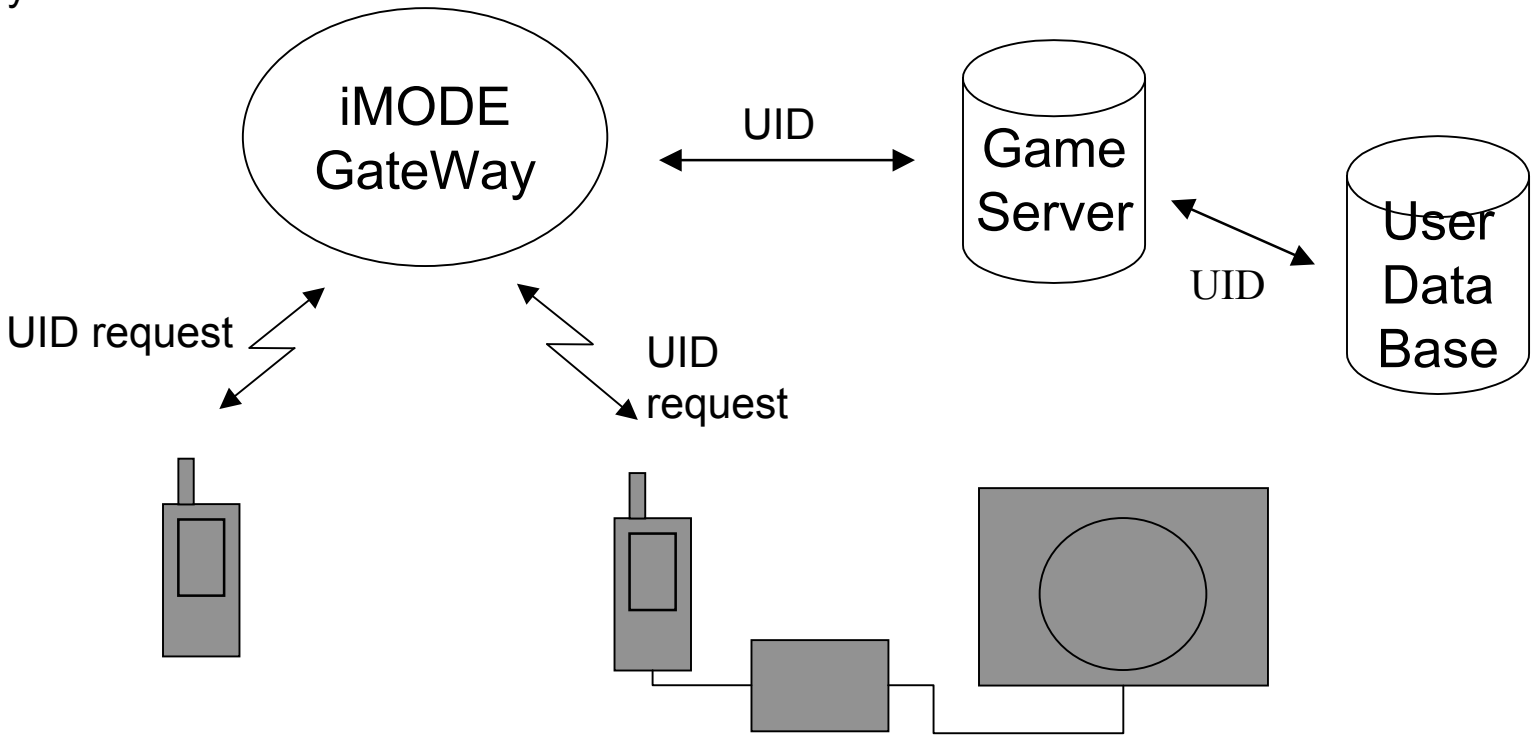
- An iMODE mobile connects the PlayStation and PlayStation 2 to the Internet.
- Because iMODE is packet-based, it creates a feeling of "always connected to the net."
- As long as the iMODE contract is completed, it is not necessary to configure or contract anew.
- The authentication and accounting system of iMODE can be used.



Concept 2: Linking With Content of an iMODE Mobile

- Data from a PlayStation, or PlayStation 2 game which is in-progress is put on the net, and both players can be synchronized by accessing this data in iMODE mobile content.

- When accessed from the PlayStation, PlayStation 2, and a mobile iMODE, the same terminal ID (UID) is sent to the server so it is easy to synchronize user data from each player.



iMODE Cable

- **Cable**
  - Connected to the controller terminals
  - Compatible with PlayStation (including PSOne) and PlayStation 2.
  - A multitap cannot already be installed.
  - Compatible only with the iMODE mobile phone that supports "iNavilink". Currently, the P502i, F502it and N502it are the only compatible equipment. The 503 series will be sold after this winter and all units will be compatible.

iMODE Cable Software

The software that goes with the cable is initially planned only for the PlayStation. The PlayStation 2 will be supported with compatibility mode.

- **Browser**
  - Compact HTML which is based on iMODE.
  - Local (CD-ROM) files can be accessed.
  - K-JAVA will not be supported.

- **Mailer**
  - Can edit/send/receive iMODE mail.

- **Telephone directory and mail editing**
  - Functions to edit the telephone directory and mail from the mobile phone.

iMODE Library (libimode)

Libraries will be provided for both PlayStation and PlayStation 2 (native).

- Organized as APIs for initialization, exit, and protocol processing.

- It is necessary to call the protocol processing API at a moderate rate | (within several V)

- The library puts received data in a receive buffer specified by the application, and sends data from a specified send buffer. The buffer contents consist of a block of HTTP.

- Communications settings such as the buffer address are set in a structure passed to the library API.

- The communications state (completed, error, etc.) is verified by the status of a structure which is passed to the library API. Status is updated whenever the protocol processing API completes.

API

Initialize library
int sceImodeInitialize (void)

Exit library
int sceImodeFinalize (void)

Protocol processing
int sceImodeProc (sceImode_Param* param)

Context Structure

```
typedef struct {
    short  fCommandType;   Command type
    short  fPortNo;        iMODE cable connection port no.(0:port1, 1:port2)
    short  fSlotNo;        iMODE cable connection slot no.(A~D:0~3)
    char*  fSendBuff;      iMODE cable connection slot no.(A~D:0~3)
    short  fSendBuffSize;  Send data buffer size
    short  fSendDataSize;  Actual size of send data
    char*  fRcvBuff;       Receive data buffer size
    short  fRcvBuffSize;   Size of receive data buffer
    short  fRcvDataSize;   Size of receive data buffer
    char*  fInfoBuff;      Info data receive buffer address
    short  fInfoBuffSize;  Info data receive buffer size
    short  fInfoDataSize;  Actual size of info data received
    short  fStatus;        State variable
    short  fLastError;     Error class
} sceImode_Param;
```

Commands

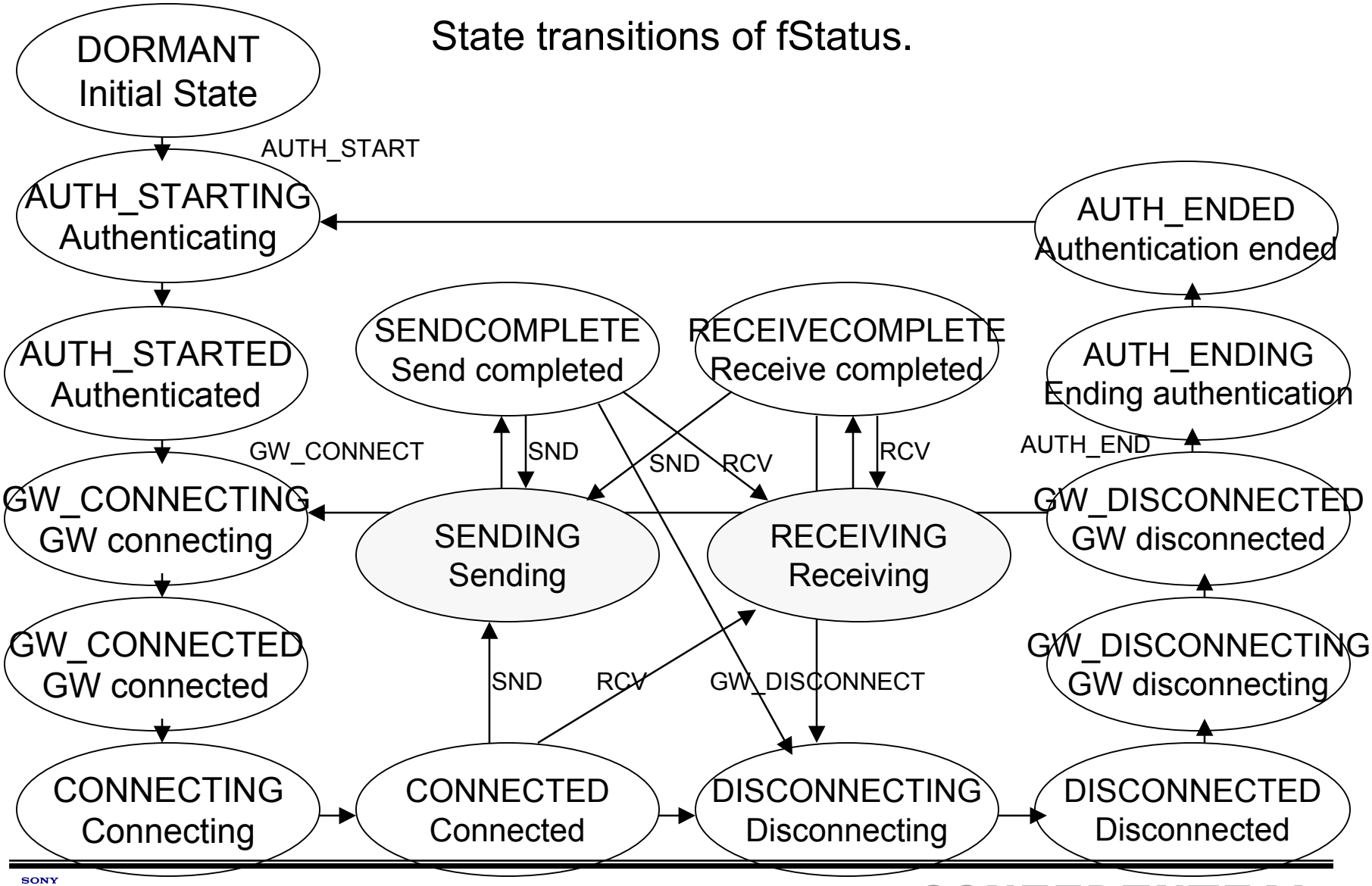
Perform processing by setting the commands below to fCommandType, then call sceImodeProc() periodically.

IMODE_OK is returned when processing completes.

Value	Symbol	Meaning
0	IMODE_CMD_RCV	Check receive data (default)
1	IMODE_CMD_SND	Send request
2	IMODE_CMD_STS	Get state
3	IMODE_CMD_ABORT	Abort send/receive
4	IMODE_CMD_AUTH_START	Begin authentication
5	IMODE_CMD_AUTH_END	End authentication
6	IMODE_CMD_GW_CONNECT	Connect gateway
7	IMODE_CMD_GW_DISCONNECT	Disconnect gateway

State Transition Diagram

State transitions of fStatus.



What is Possible? (1)

- **HTTP data: Get, Post, Head**
  - The communications library that is provided operates at the HTTP level and sends and receives blocks of HTTP (consisting of status / header / body) (In other words, arbitrary data can be exchanged with an HTTP server.)
  - In the body, arbitrary data can be received, but send data must be in TEXT format.
  - Details of the protocol are published separately.

- **Push (distributed communications)**
  - HTTP data from the server can be pushed without the client making a request.
  - Server events can be initiated asynchronously without polling the server.

- **Accounting & Authentication**
  - The iMODE terminal authentication system can be used. For more information, see the description of "UID" under "Server".
  - The iMODE accounting agent can be used. For more information, see the description of "Accounting" under "Server".
  - These features can be used optionally.

What is Possible? (2)

- **Communicating with the CPU and memory of a mobile phone**
  - We also plan to provide a low-level API for communicating with the CPU of a mobile phone. With this API, functions such as download and upload of an Address Book will allow an individual mobile phone to be accessed. However, because of differences between manufacturers, only the lowest level of communication will be supported for these functions.

What is NOT Possible?

- **Communication using protocols other than HTTP**
  - The library that is provided communicates at the HTTP level using a low layer protocol, therefore, applications that use application protocols other than HTTP (e.g. SMTP, FTP) cannot be used.

- **K-JAVA**
  - There are plans to provide a K-JAVA VM that runs on the PlayStation and PlayStation 2 but it is currently not available.
  - It is possible to communicate with a K-JAVA VM running on a mobile phone. (HTTP protocol via the server)

Server (1)

- **Server software**
  - Ordinary HTTP server software (e.g. Apache, Netscape Enterprise) can be used.
  - The protocol is basically HTTP, but, there are extensions for iMODE and parts are restricted. Details are provided separately.

- **UID**
  - For sites that have been formally registered with DoCoMo, the accessing user ID (UID) can be obtained. One UID corresponds to one mobile phone. This is the DoCoMo accounting agent's unit.
  - Using the UID, private user data can be managed. Requiring ID and password to be input is not a problem.
  - The UID is same when connecting with the iMODE unit, and when connecting via the PlayStation or the PlayStation 2. Consequently, game data from both sides can be linked.
  - How to perform formal registration of a site and obtain the UID is described separately.

Server (2)

- **User agent**
  - When connecting to the server from the PlayStation or PlayStation 2, in order to recognize this connection, it is necessary to specify the private user agent in the request header.
  - There are plans to include "Vendor ID" and "Title ID" information in the user agent when making a connection from the PlayStation or PlayStation 2. Details provided separately.

- **Accounting**
  - For sites formally registered with DoCoMo, the DoCoMo accounting agent service can be used. An account costs 100 to 300 yen per month. DoCoMo collects the payment together with the telephone fee, and the content provider receives a commission.
  - It is not necessary to use this accounting system, and other means of charging are acceptable.

Design Precautions (1)

- **iMODE packet features**
  - When the packet charge paid by the user becomes too high, the user will disconnect. Both the design and the technology should allow the number of skipped packets that are requested to decrease as much as possible.
  - As much as possible, required content should be put on CD. We recommend limiting the content to be downloaded to only that which is absolutely necessary.

- **Accessing many unspecified sites**
  - (Inputting a URL for the user.) Please contact us separately if you plan to create an application that accesses many unspecified sites. The authentication procedure of NTT DoCoMo may be required.

Design Precautions (2)

- **Guidelines exist for specific operations**
  - Although they are explained separately, it is necessary to create My Menu-related operations according to the operating procedure and messages specified here.
- **Consider the response**
  - In iMODE-based communications, there normally is a response within several seconds. However, when the broadcast signal is bad, or the Center is having trouble, responses may take longer which can lead to a timeout. Try to avoid confusing the user in these cases.

Technical Precautions

- **It is necessary to ensure that communication is performed every 1V**
  - When AUTH_START is executed with sceProc(), communication with the iMODE cable begins in interrupt mode every 1V, and completes with AUTH_END. Be careful not to hinder this communication. Specifically, do not perform processing in user interrupt mode that exceeds 1V (e.g. VSyncCallback), and do not prohibit interrupts for long periods (Critical Section).
  - If 1V operation cannot be ensured, communication may fail. If communication stops for more than 50 msec, the iMODE cable microprocessor will reset.

  (For PS2, this discussion concerns the IOP)

- **Consider communication failures**
  - For starters, the Internet is unstable. Consequently, iMODE communication will not always be stable. Applications should be designed to recover in case of communication failures.

- **Consider the arrival of voice and mail messages**
  - Voice and mail messages can arrive during data communication. Because that state can arise from the library, it is necessary to perform appropriate processing. Guidelines are provided separately.
  - Basically, it should be such that mail should not be taken from the Center while playing a game. However, even in that case, it is necessary to notify the user of arriving communication.

Q&A