第八周实习报告

5080309684 仇卓

任务概述

继续改进代码，修改一些 bug，完善服务层管理。

实现要点

服务器端更新设备信息时，完善服务类型等信息

```
public void start(BundleContext bundleContext) throws Exception {
    Activator.context = bundleContext;
    //update();
    ServiceReference[] dvs = bundleContext.getServiceReferences(
            UPnPDevice.class.getName(),
            "(ObjectClass=" + UPnPDevice.class.getName() + ")");

    if (dvs == null) {
        System.out.println("No UPnP device found");
    } else {
        for (int i = 0; i < dvs.length; i++) {
            Hashtable info = new Hashtable();
            String[] keys = dvs[i].getPropertyKeys();
            //System.out.println(Arrays.toString(keys));
            for (int j = 0; j < keys.length; j++) {
                info.put(keys[j], dvs[i].getProperty(keys[j]));
            }
            System.out.println("Device: " + dvs[i].getProperty(UPnPDevice.UDN));

            UPnPDevice dev = (UPnPDevice) context.getService(dvs[i]);
            UPnPService[] services = dev.getServices();
            Hashtable svs = new Hashtable(services.length);
            for (int j = 0; j < services.length; j++) {
                UPnPAction[] actions = services[j].getActions();
                ArrayList acns = new ArrayList(actions.length);

                // 加入服务类型信息，便于服务层管理
                acns.add(services[j].getType());
                for (int k = 0; k < actions.length; k++) {
                    ArrayList parameters = new ArrayList(3);
                    parameters.add(actions[k].getName());
```

```
                        parameters.add(actions[k].getInputArgumentNames());
                        parameters.add(actions[k].getOutputArgumentNames());
                        acns.add(parameters);
                    }
                    svs.put(services[j].getId(), acns);
                }
                info.put("Device Services", svs);
                devices.add(info);
            }
        }

        bundleContext.addServiceListener(this,
                "(ObjectClass=" + UPnPDevice.class.getName() + ")");

        Hashtable tmp = new Hashtable();
        Filter fi = null;
        try {
            fi = bundleContext.createFilter(filter);
        } catch (Exception e) {
            e.printStackTrace();
        }
        tmp.put(UPNP_FILTER, fi);
        srr = bundleContext.registerService(
                UPnPEventListener.class.getName(), this, null);

        ss = new ServerSocket(22222);
        thread.start();
    }

    private void update() throws Exception {
        ServiceReference[] dvs = context.getServiceReferences(
                UPnPDevice.class.getName(),
                "(ObjectClass=" + UPnPDevice.class.getName() + ")");
        devices.clear();

        if (dvs == null) {
            System.out.println("No UPnP device found");
        } else {
            for (int i = 0; i < dvs.length; i++) {
                Hashtable info = new Hashtable();
                String[] keys = dvs[i].getPropertyKeys();
                for (int j = 0; j < keys.length; j++) {
                    info.put(keys[j], dvs[i].getProperty(keys[j]));
                }
```

```
                UPnPDevice dev = (UPnPDevice) context.getService(dvs[i]);
                UPnPService[] services = dev.getServices();
                Hashtable svs = new Hashtable(services.length);
                for (int j = 0; j < services.length; j++) {
                    UPnPAction[] actions = services[j].getActions();
                    ArrayList acns = new ArrayList(actions.length);
                    // 加入服务类型信息，便于服务层管理
                    acns.add(services[j].getType());
                    for (int k = 0; k < actions.length; k++) {
                        ArrayList parameters = new ArrayList(3);
                        parameters.add(actions[k].getName());
                        parameters.add(actions[k].getInputArgumentNames());
                        parameters.add(actions[k].getOutputArgumentNames());
                        acns.add(parameters);
                    }
                    svs.put(services[j].getId(), acns);
                }
                info.put("Device Services", svs);
                devices.add(info);
            }
        }
    }
```

客户端

客户端显示服务时，按照服务类型将服务分类放入不同的容器，然后将相同服务类型的所有服务显示在一起。

注意事项和问题

虽然程序实现了服务层管理，用户可以看到不同的服务类型，每个服务类型有不同的服务，用户看不到服务的所属设备。但是 UPnP 网络似乎非常智能，对于名称相同的服务类型或名称，UPnP 网络会自动修改服务类型或名称，在后面加入一个数字后缀等，可以通过比较前缀来归类同类型的服务。

参考资料

http://dz.prosyst.com/pdoc/mbserver_5.2/um/upnp/developer/osgi_upnp/osgi_upnp_export.html
Writing a UPnP™ Device with the OSGi API

http://dz.prosyst.com/pdoc/mbserver_5.1/Tutorial/upnp/bundles/upnp/upnp.html
UPnP Driver Bundle