Putting a SOUL in software engineering tools
An overview of logic program querying and its applications
package com.bt.sdp.onboarding.controllers.service.lifecycle;

import java.util.Map;

/**
 * @author Tim Watson
 */

@Controller
public class PublicationController {

    private final ServiceStorageGateway gateway;

    @Autowired
    public PublicationController(final ServiceStorageGateway gateway) {
        this.gateway = gateway;
    }

    @SuppressWarnings("unchecked")
    @RequestMapping(value = "/private/service/publish", method = RequestMethod.POST)
    public final String publishService(final Map model) {
        DeploymentStage currentStage = gateway.getDeploymentStageByServiceId(id);
        if (currentStage == DeploymentStage.Deployed || currentStage == DeploymentStage.Uploaded) {
            gateway.updateDeploymentStage(id, DeploymentStage.Published);
            model.put(ViewConstants.XML_DATA_KEY, ViewConstants.statusMessage("Service is ready to be published.
            It must be in the Published state.
            ");
        } else {
            model.put(ViewConstants.XML_DATA_KEY, ViewConstants.statusMessage("Service is not ready to be published.
            It must be in Deployed state.
            ");
        }
        return "publishServiceResponse";
    }

    @RequestMapping(value = "/private/service/unpublish", method = RequestMethod.POST)
    public final String unpublishService(final Map model) {
        if (gateway.getDeploymentStageByServiceId(id) == DeploymentStage.Published) {
            gateway.updateDeploymentStage(id, DeploymentStage.Unpublished);
            model.put(ViewConstants.XML_DATA_KEY, ViewConstants.statusMessage("Service is not ready to be unpublished.
            Only unpublished Service can be unpublished.
            ");
        } else {
            model.put(ViewConstants.XML_DATA_KEY, ViewConstants.statusMessage("Service is already unpublished.
            ");
        }
        return "unpublishServiceResponse";
    }
}
if ? class in Class1 {Method1(); return;

? method in Method1{ other; end;

? method methodWithVme? arg-1;
HOW TO DISASSEMBLE

1. [Diagram of step 1]
2. [Diagram of step 2]
3. [Diagram of step 3]
4. [Diagram of step 4]
5. [Diagram of step 5]
6. [Diagram of step 6]

Billy

[Image of a Billy bookcase]
if jtClassDeclaration(\$classDeclaration) {
    class \$className {
        private \$fieldDeclarationType \$fieldName;
        \$modifierList \$returnType \$methodName(\$parameterList) {
            return \$fieldName;
        }
    }
}
cntExecuted(?n, ?b, ?c)
cntExited (?n, ?a)
cntEntered (?n, ?a)