

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики
Факультет «Информационных технологий и программирования»

А. А. Борисенко, Д. М. Пенькин

Моделирование работы автомобильной сигнализации
с помощью инструментального средства *Unimod*

Проект создан в рамках
«Движения за открытую проектную документацию»

Санкт-Петербург
2008

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ПОСТАНОВКА ЗАДАЧИ	4
2. СЦЕНАРИИ РАБОТЫ	6
2.1. НЕФОРМАЛЬНОЕ ОПИСАНИЕ	6
2.1.1. <i>Охранная функция системы</i>	6
2.1.2. <i>Функции системы при включенном зажигании</i>	7
2.1.3. <i>Программирование системы</i>	7
2.2. ФОРМАЛЬНОЕ ОПИСАНИЕ	7
2.2.1 <i>Описание связи автоматов с источниками событий и объектами управления</i>	8
2.2.2 <i>Описание автомата, отвечающего за эмулирование физического состояния автомобиля</i>	10
2.2.3 <i>Описание автомата, отвечающего за работу сигнализационной системы автомобиля</i>	10
2.2.4 <i>Описание автомата, отвечающего за работу пульта дистанционного управления</i> ..	11
2.3. ПРИМЕРЫ ВНЕШНЕГО ВИДА МОДЕЛИ СИГНАЛИЗАЦИОННОЙ СИСТЕМЫ	12
С ЧЕГО ВСЕ НАЧИНАЛОСЬ	14
3. ИСПОЛНЕНИЕ ПРОГРАММЫ	16
3.1 ИНТЕРПРЕТАЦИОННЫЙ ПОДХОД	16
3.2 КОМПИЛЯТИВНЫЙ ПОДХОД	16
ВЫВОДЫ	17
ИСТОЧНИКИ	18
ПРИЛОЖЕНИЯ	19
Приложение 1. ОПИСАНИЕ JAVA-КЛАССОВ	19
1.1. IREMOTEGUI.JAVA	19
1.2. IREMOTECONTROL.JAVA	19
1.3. ICAR.JAVA	19
1.4. MOUSEAREAWATCHER.JAVA	20
1.5. CARPANEL.JAVA	20
1.6. REMOTECONTROLPANEL.JAVA	23
1.7. REMOTECONTROLBACKPANEL.JAVA	24
1.8. SIREN.JAVA	27
Приложение 2. XML-ОПИСАНИЕ МОДЕЛИ	29
2.1. A1CAR.XML	29
2.2. A2SYSTEM.XML	37
2.3. A3REMOTE.XML	44

ВВЕДЕНИЕ

На сегодняшний день человек, живущий в городе, во многом полагается на различные технические устройства. Со временем их становится все больше и больше, поскольку они способны решать множество задач, и их можно «научить» решать новые задачи, так как большинство таких устройств являются программируемыми. Очевидно, что массовость и повсеместность использования вкупе с постоянной потребностью разработки новых устройств требуют задуматься об оптимизации процесса проектирования, реализации и отладки программного обеспечения для таких устройств.

Основными факторами, определяющими эффективность подхода к проектированию устройства, являются простота и надежность, поскольку именно в таком случае разработчикам удастся избежать наибольшее число ошибок, что существенно облегчит как отладку, так и использование устройства конечными пользователями. Возможность *формализации* подхода, а значит, возможность на каком-то уровне доказать корректность реализуемых алгоритмов, могла бы существенно увеличить его надежность. Однако в общем случае формализация и доказательство – это достаточно трудоемкие процессы, поэтому возникает вопрос о возможности автоматизации всего процесса создания программы. Такой подход позволит сократить затраты разработчиков, а главное, обеспечит автоматический контроль качества на базовом уровне.

В данном проекте предлагается использовать автоматный подход [1–4], который в полной мере удовлетворяет рассмотренным требованиям. Идея такого метода заключается в том, что при разработке программы от человека требуется лишь проектирование и создание формальной спецификации, представляющей собой систему конечных автоматов, по которой затем *автоматически* генерируется конкретная реализация системы на некотором языке программирования. В данном проекте приводится также пример разработанного по рассмотренному методу устройства.

В данном проекте на основе автоматного подхода реализовано программное обеспечение для эмуляции автомобильной сигнализационной системы. В качестве инструментального средства использован *Unimod* [1, 2], генерирующий код на языке *Java*.

Необходимо отметить, что уже существуют варианты решения данной задачи, выполненные по схожему методу, например [5], однако данный проект расширяет возможности сигнализационной системы.

Глава 1. ПОСТАНОВКА ЗАДАЧИ

Под автомобильной сигнализационной системой будем понимать совокупность, собственно, охранного блока, устанавливаемого в автомобиле, и пульта управления, находящегося у владельца автомобиля. Связь между охранным блоком и пультом управления осуществляется через радиоканал. Охранный блок – это автоматизированное устройство, позволяющее автономно выполнять следующие функции:

- при нарушении охраняемых зон в режиме охраны:
 - всевозможными способами сообщить об этом владельцу;
 - привлечь внимание окружающих с помощью световых и звуковых сигналов с целью отпугнуть возможного взломщика;
 - предпринять действия по восстановке нарушенных зон охраны;
- сообщать о состоянии системы на пульт управления;
- выполнять команды, поступающие с пульта управления:
 - запирать и отпирать двери автомобиля (при включенном зажигании);
 - включать и выключать режим охраны;
 - выключать тревогу;
 - включать режим паники;
 - изменять параметры системы (режим программирования системы).

Большинство действий системы подтверждается звуковыми и световыми сигналами, а также уведомлениями на пульт управления, что облегчает взаимодействие пользователя с сигнализационной системой.

Программа реализует три подсистемы: алгоритм работы охранного блока сигнализационной системы, алгоритм работы пульта управления, а также протокол обмена данными между ними.

Взаимодействие пользователя с системой подразумевает использование только пульта управления. Следует рассмотреть реальную ситуацию на сегодняшний день: подавляющее большинство автомобильных сигнализационных систем, имеющих возможность настройки параметров, предполагают их изменение через включение/выключение зажигания при открытой водительской двери по описанным в инструкции схемам. Например, для изменения чувствительности датчика удара необходимо:

- включить режим настройки системы, выключив режим охраны, открыв водительскую дверь и включив и выключив зажигание восемь раз подряд;
- дождаться подтверждения включения режима настройки системы быстрым миганием светодиода;
- выбрать настройку датчика удара, включив и выключив зажигание три раза подряд;
- дождаться подтверждения включения режима настройки датчика удара быстрым миганием светодиода;
- включить и выключить зажигание от одного до пяти раз подряд, в зависимости от желаемой чувствительности датчика (1 – низкая, 5 – высокая чувствительность).

На взгляд авторов, данный алгоритм крайне неудобен для пользователя, более того,

он ограничивает возможность расширения функций сигнализационной системы. В связи с этим, предлагается использовать пульт управления, имеющего интуитивный графический пользовательский интерфейс, сходный с интерфейсом мобильного телефона. Это позволит сделать все функции настройки системы гораздо удобнее и привычнее для пользователя. Набор настроек реальной автомобильной сигнализации может быть весьма велик, но в большинстве своем он рассчитан на подстройку под конкретный автомобиль. Например, одной из необходимых настроек системы является длительность электрического импульса, необходимого для нормального функционирования активаторов дверных замков (если активатор электрический, то достаточно и половины секунды, а если пневматический, то длительность импульса равна паре секунд). Данный проект не учитывает подобные тонкости, однако наглядно показывает, насколько удобно может быть осуществлена, по сути, любая настройка сигнализационной системы.

Глава 2. СЦЕНАРИИ РАБОТЫ

2.1. Неформальное описание

Основная задача системы заключается в выполнении охранной функции, а именно, в оповещении владельца автомобиля о нарушении охраняемых зон (открытие двери или замка, срабатывание датчика удара) и в привлечении внимания окружающих с помощью световых и звуковых сигналов. Система способна управлять состоянием замков автомобиля, а также блокировать двигатель автомобиля разрывом одной из обеспечивающих его работу устройств, например, обесточиванием электрического бензонасоса. Помимо этого, при включенном зажигании сигнализация дублирует функции центрального замка автомобиля, то есть позволяет открывать и закрывать замки дверей автомобиля.

Основной блок системы устанавливается в автомобиле, который управляется пользователем с помощью пульта дистанционного управления. Пульт связывается с основным блоком по радиоканалу, все команды шифруются для защиты от несанкционированного управления системой путем перехвата команд в радиоэфире.

Система функционирует постоянно, и ее функции зависят, в частности, от состояния автомобиля. Например, если зажигание включено, то она управляет лишь центральным замком, запирая или отпирая замки дверей по желанию пользователя. При выключенном же зажигании пользователь имеет возможность включать или выключать режим охраны, а также настраивать систему.

2.1.1. Охранная функция системы

Пользователь может включить режим охраны только при выключенном зажигании. В этом случае система блокирует двигатель, запирает замки дверей, капота, багажника и бензобака, а также закрывает стекла. Затем включается датчик удара. В подтверждение выполнения этих операций система подает звуковой и световой сигналы. Если по каким-либо причинам операции не были полностью выполнены – например, пользователь не закрыл одну из дверей, – система сообщит об этом, а проблемная зона будет исключена из списка охраняемых до ее восстановления.

В режиме охраны система, кроме как на команды с пульта управления, реагирует на следующие события:

- *Сильный или слабый удар.* Датчик, регистрирующий удары по кузову автомобиля, является двухуровневым, что позволяет отличать слабые и сильные удары. Слабый удар вызывает лишь короткий предупредительный сигнал, в то время как сильный удар приравнивается к нарушению зоны охраны.
- *Нарушение зоны охраны.* Под термином «зона охраны» понимается в отдельности каждая дверь, крышки багажника, моторного отсека и лючок бензобака. Каждая из зон охраны имеет свой, отдельный датчик открытия.
- *Включение зажигания.* Включение зажигания в режиме охраны означает несанкционированную попытку завести двигатель, поэтому данное событие немедленно вызывает реакцию системы.

- *Использование центрального замка.* Центральный замок автомобиля обычно устанавливается еще на заводе, поэтому пользователь может взаимодействовать с ним напрямую, а не через охранную систему. В связи с этим, система должна реагировать на любые попытки несанкционированного управления центральным замком автомобиля в режиме охраны. При возникновении данного события система, помимо подачи звуковых и световых сигналов, пытается снова запереть все замки.

При выходе из режима охраны система отпирает все замки, снимает блокировку двигателя и сообщает о произошедших событиях с помощью несложной схемы световых и звуковых сигналов.

2.1.2. Функции системы при включенном зажигании

При включении зажигания система автоматически запирает замки во избежание случаев открытия двери во время движения автомобиля. Пользователю же остаются доступными лишь функции запираения и отпираения замков дверей, что фактически дублирует функции центрального замка автомобиля.

2.1.3. Программирование системы

Программирование системы осуществляется только при выключенном зажигании и только с использованием пульта управления. В число программируемых настроек и функций входят:

- чувствительность датчика удара;
- настройка режима оповещения.

Чувствительность датчика удара влияет на величину силы удара по кузову автомобиля, которую система будет воспринимать как опасность. Датчик удара системы может регистрировать сильные и слабые удары. На слабые удары система реагирует лишь предупреждением, сильные же интерпретируются как попытка взлома или угона, поэтому система переходит в режим тревоги и оповещает владельца о происшествии, посылая соответствующее сообщение на пульт дистанционного управления.

Вторая настройка касается способа оповещения владельца о состоянии автомобиля. Принимая сообщение о тревоге, пульт выводит его на экран и издает звук, если не находится в тихом режиме. В тихом режиме сообщение выводится только на экран.

Как уже было упомянуто, программирование системы осуществляется с помощью меню пульта управления. Перемещение по меню осуществляется с помощью клавиш постановки и снятия с охраны (влево и вправо соответственно), а выбор пункта меню – нажатием на центральную клавишу (клавишу «Паника»). Интерфейс меню интуитивно понятен, что облегчает настройку системы. Изменение настроек пульт подтверждает коротким сигналом, если только он не находится в тихом режиме. Автомобиль подтверждает изменение настроек, касающихся охранного блока, с помощью световых и звуковых сигналов.

2.2. Формальное описание

Для реализации сценария необходимо установить *объекты управления*, над которыми производятся операции.

Объектами управления являются системы автомобиля: устройства подачи звуковых и световых сигналов, замки дверей, капота, багажника и бензобака, система блокировки двигателя. Помимо этого, пульт дистанционного управления также является объектом управления.

Управление перечисленными объектами производится с помощью событий, инициируемых *источниками событий*, которыми являются: пульт дистанционного управления, датчики охраняемых зон, датчик удара, замок зажигания. События обрабатываются системой автоматов, взаимодействующих между собой. Формальное описание, содержащее в себе объекты управления, источники событий и систему автоматов, приведено ниже.

Для пульта объектами управления являются блок сигнализации в автомобиле и система отправки сообщений охранному блоку. Источниками событий для пульта являются блок сигнализации и физические кнопки пульта.

2.2.1 Описание связи автоматов с источниками событий и объектами управления

В системе представлены три конечных автомата: один отвечает за регистрацию физического состояния автомобиля, второй представляет схему работы сигнализационной системы, а третий – пульта дистанционного управления. С точки зрения инструментального средства *Unimod*, автоматы соединены с источниками событий и объектами управления так, как показано на рис. 1.

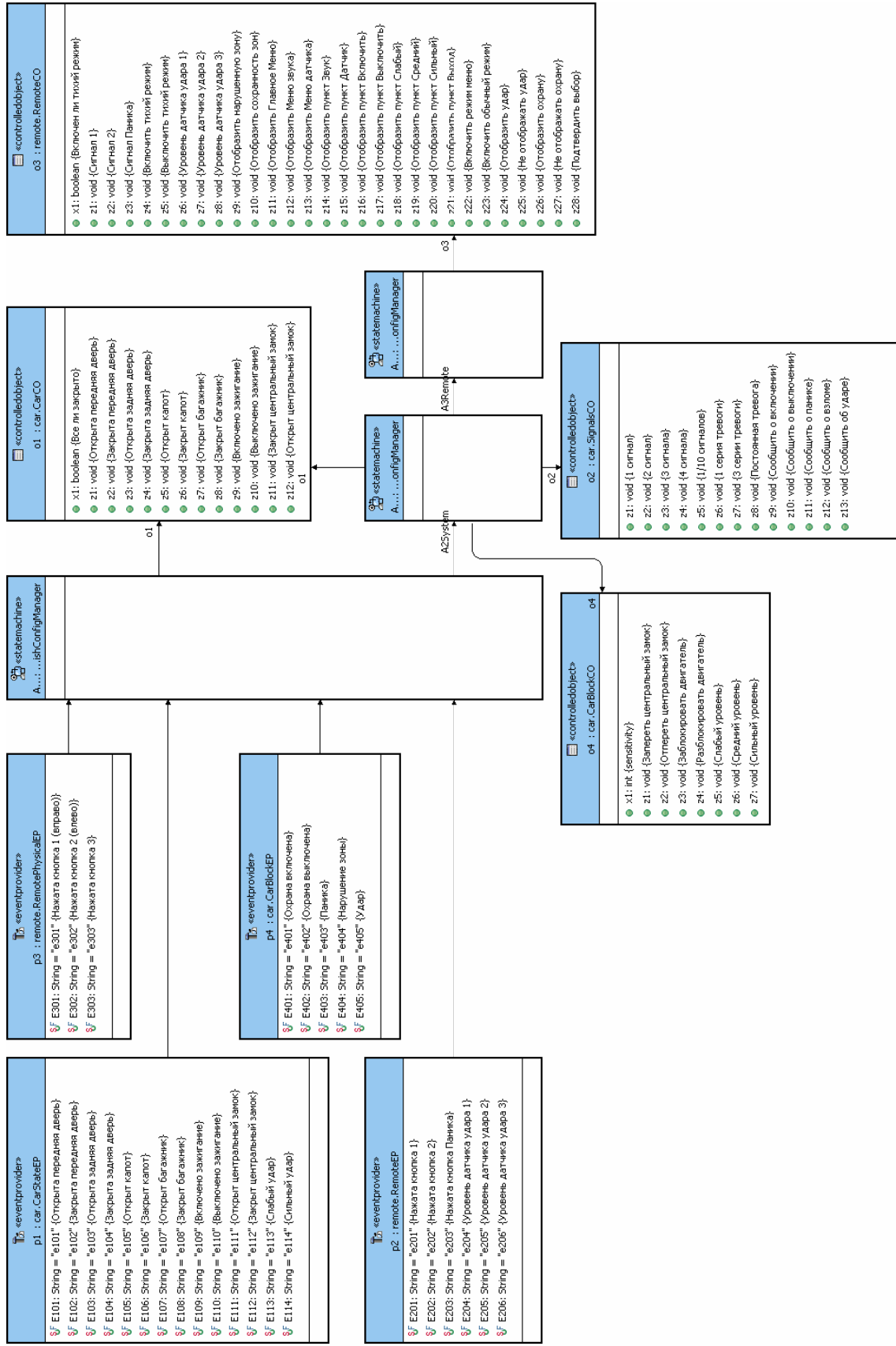


Рис. 1 Описание связи автоматов с источниками событий и объектами управления

2.2.2 Описание автомата, отвечающего за эмуляцию физического состояния автомобиля

Один из автоматов системы служит для эмуляции физического состояния автомобиля: он регистрирует открывание дверей, капота, багажника и включение зажигания. Поскольку это единственная его задача, устроен он очень просто (рис. 2).

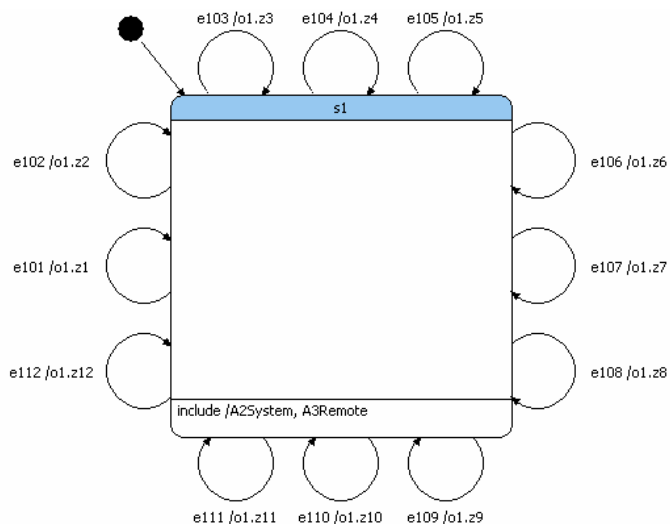


Рис. 2 Описание автомата, отвечающего за эмуляцию физического состояния автомобиля

2.2.3 Описание автомата, отвечающего за работу сигнализационной системы автомобиля

Данная схема (рис.3) описывает состояния сигнализационной системы. Поскольку система работает постоянно, в автомате нет допускающего состояния, означающего конец работы.

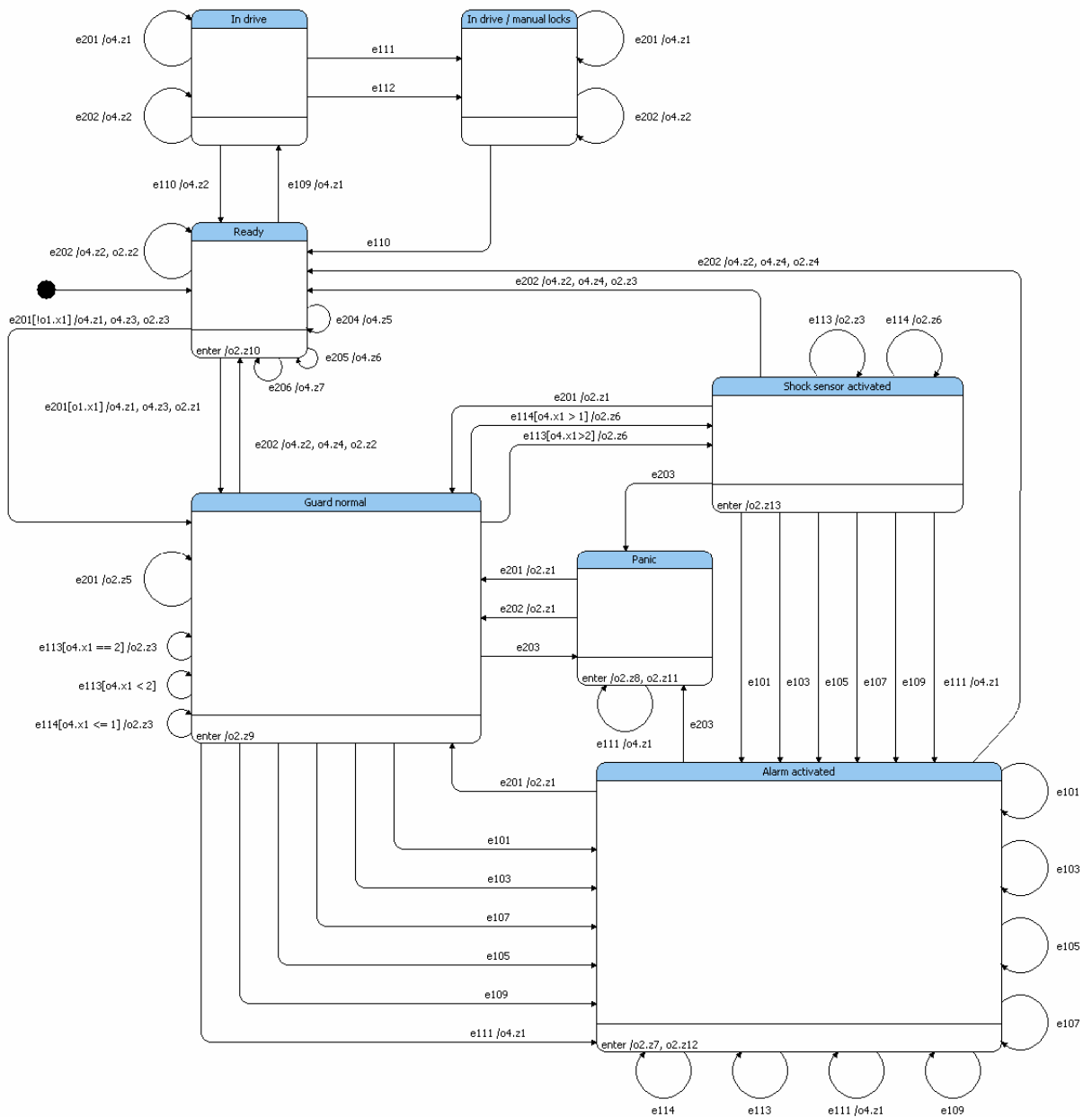


Рис. 3 Описание автомата, отвечающего за работу системы, установленной в автомобиле

2.2.4 Описание автомата, отвечающего за работу пульта дистанционного управления

Схема на рис. 4 описывает состояния пульта дистанционного управления системой. В схеме явно выделяется «дерево меню», которое, как видно, может быть дополнено любым количеством опций с любой степенью вложенности. Поскольку система работает постоянно, в автомате нет допускающего состояния, означающего конец работы.

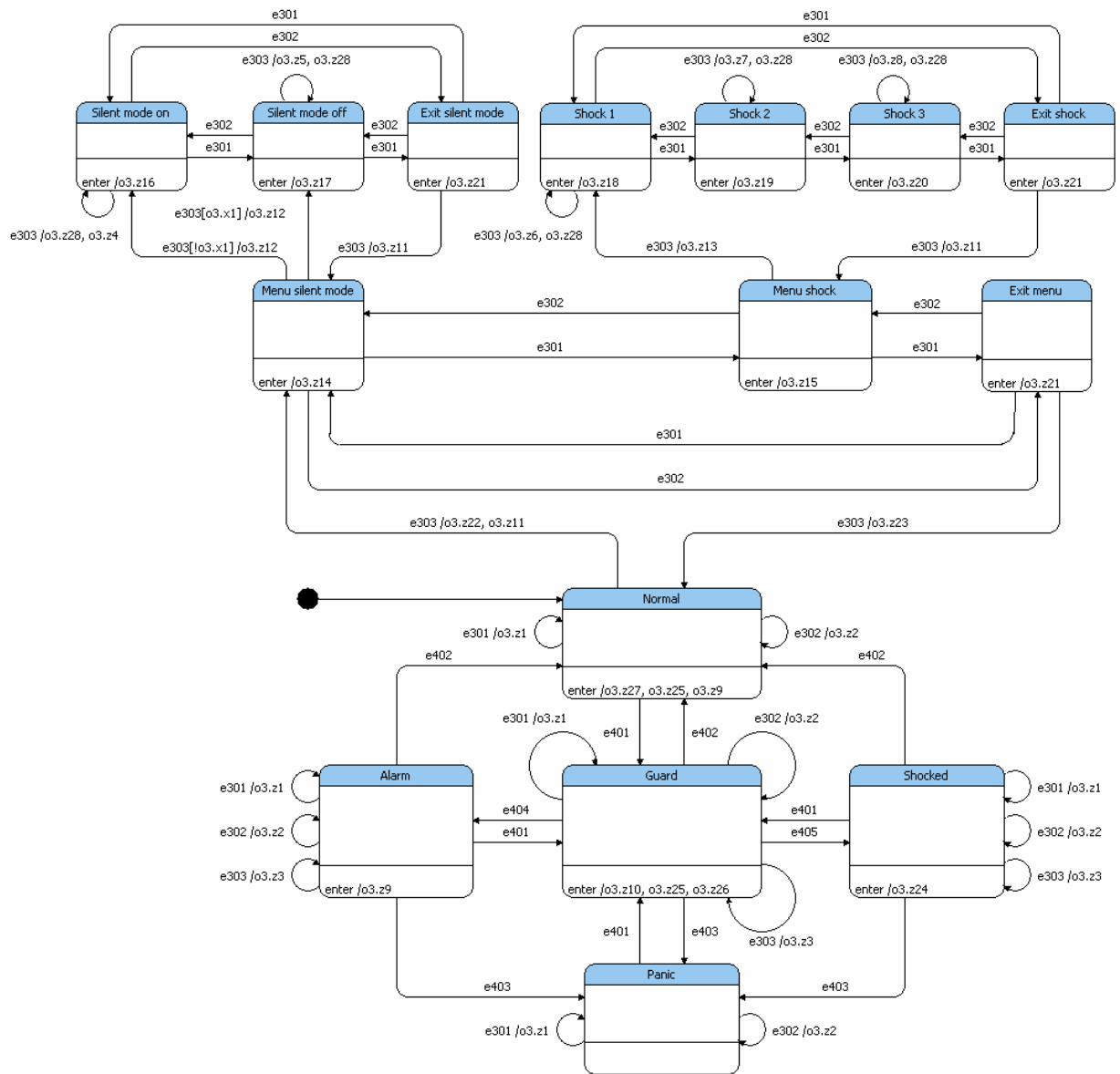


Рис. 4 Описание автомата, отвечающего за работу пульта дистанционного управления

2.3. Примеры внешнего вида модели сигнализационной системы

В этом разделе будут приведены виды интерфейса эмулятора автомобильной сигнализационной системы. Интерфейс состоит из двух окон, одно из которых отводится автомобилю, другое – пульту дистанционного управления. Пульт намеренно отображен с двух сторон. Идея авторов заключалась в достижении минимальных размеров пульта управления системой. В связи с этим было принято решение разместить экран устройства на его обратной стороне. За основу пульта управления был взят реально существующий ключ-брелок от автомобиля *Audi*.

Вид эмулятора в начальный момент времени приведен на рис.5.



Рис. 5 Окно отображения автомобиля

На экране пульта дистанционного управления в режиме охраны и режиме ожидания отображаются четыре пиктограммы:

- *Динамик или перечеркнутый динамик.* Эта пиктограмма отображает режим работы пульта: обычный или тихий. В тихом режиме пульт не подает никаких звуковых сигналов.
- *Открытый или закрытый замок.* Замок отображает текущее состояние охранной системы: закрытый замок означает ее включение, открытый – выключение.
- *Символ молотка.* Этот символ появляется только в случае, если система зарегистрировала сильный удар (который расценивается как попытка взлома или угона). В противном случае на месте этой пиктограммы ничего нет.
- *Открытая или закрытая дверь.* Если охрана включена, дверь отображается закрытой. Пиктограмма меняется на открытую дверь в том случае, если охранная система зарегистрировала нарушение какой-либо зоны охраны. Если охрана выключена, дверь всегда отображается открытой.

Общий вид пульта дистанционного управления представлен на рис. 6.



Рис. 6 Окно отображения пульта дистанционного управления

С помощью активных элементов на картинке, кнопок на панели управления и кнопок на пульте дистанционного управления пользователь может управлять состоянием автомобиля. Все возникающие события заносятся в журнал событий. Вид интерфейса в один из моментов работы программы показан на рис. 7.



Рис. 7 Вид интерфейса в один из моментов работы программы

В режиме программирования на экране пульта дистанционного управления отображается меню. Один из видов меню приведен на рис.8.

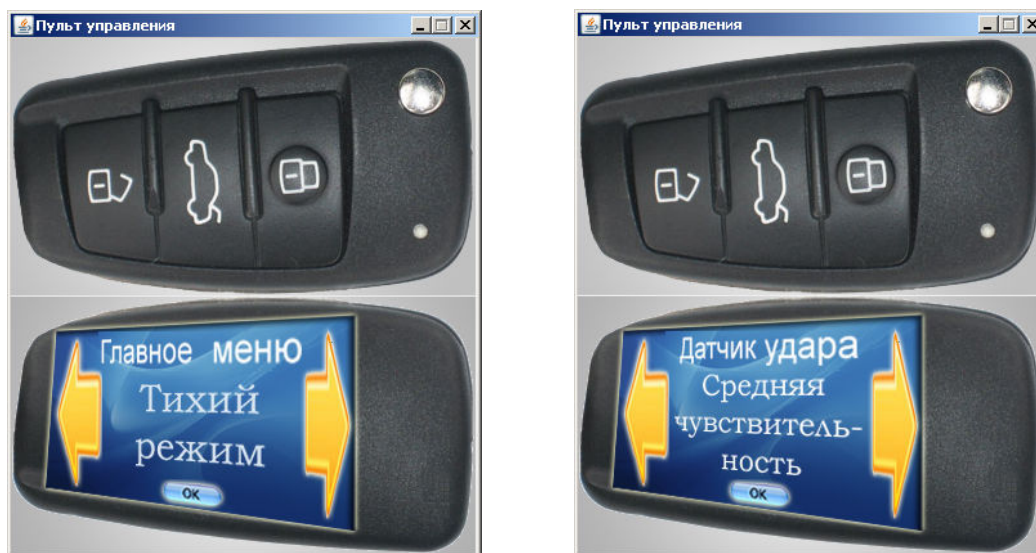


Рис. 8 Вид пульта дистанционного управления в режиме программирования

С ЧЕГО ВСЕ НАЧИНАЛОСЬ...

Вы думаете, все картинки взяты из сети Интернет? Нет, авторы проекта целый день занимались натурными съемками автомобиля и записью звуковых сигналов, а затем производили отбор и обработку отснятого материала. Все материалы прилагаются к настоящей документации и выложены на сайте по автоматному программированию и мотивации к творчеству (<http://is.ifmo.ru/>) в разделе «Проекты». Приблизительный объем только отснятых фотографий приведен на рис.9.

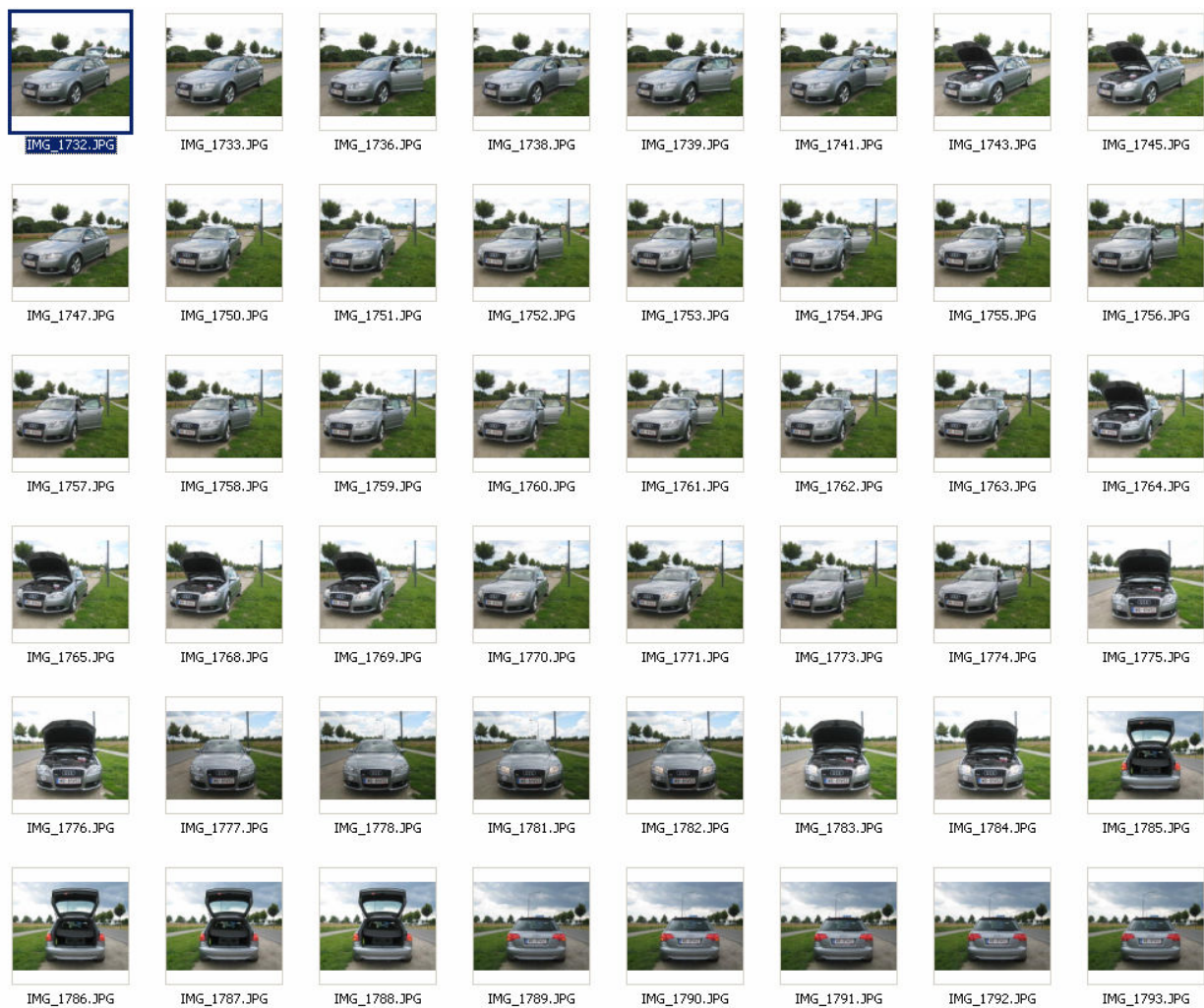


Рис. 9. Исходный материал для проекта

Глава 3. ИСПОЛНЕНИЕ ПРОГРАММЫ

Использованное в ходе разработки программы инструментальное средство *Unimod* поддерживает два подхода к исполнению программ: интерпретационный и компилятивный.

3.1 Интерпретационный подход

При интерпретационном подходе автоматы запускаются, как исходный код – используется *XML*-описание модели (Приложение 1), которое интерпретируется *java*-классом. Для работы интерпретатору необходим *jar*-файл (архив) с проектом и библиотекой *Unimod* (рис. 8).

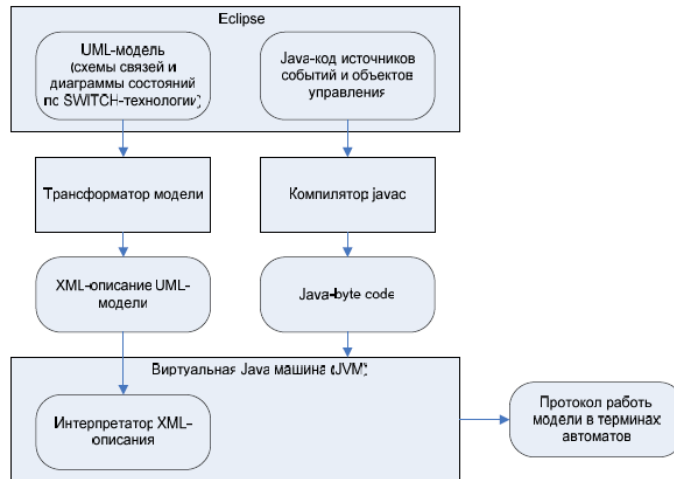


Рис.8. Схема интерпретационного подхода

3.2 Компилятивный подход

При компилятивном подходе *XML*-описание транслируется в *Java*-класс (Приложение 2). Этот класс и классы, реализующие источники событий и объекты управления, компилируются в единое *Java*-приложение, не зависящее от средства *Unimod*. Для исполнения скомпилированной программы потребуется библиотека, поставляемая вместе с инструментальным средством *Unimod* (рис. 9).

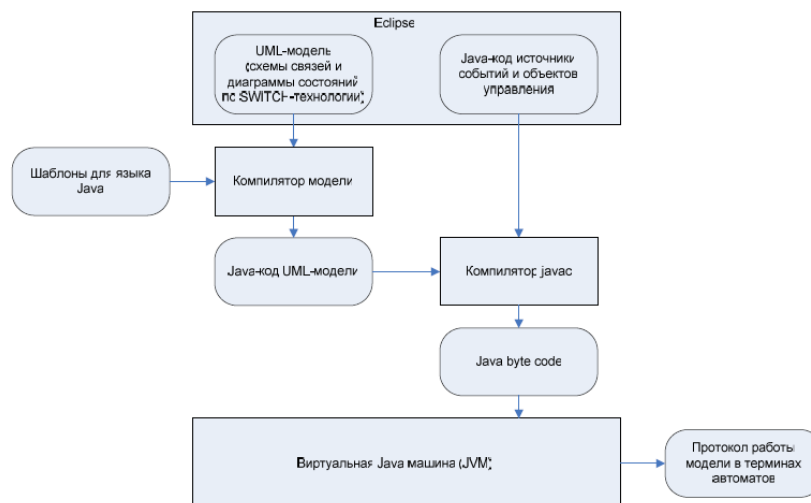


Рис.9. Схема компилятивного подхода

ВЫВОДЫ

Из данной работы следует, что использование автоматного подхода при разработке программ позволяет еще на этапе проектирования избежать существенных ошибок, давая возможность четко и наглядно представить логику программы. Отметим также и значительное улучшение документации по сравнению с традиционным подходом, сокращающее затраты на освоение того или иного программного продукта сторонними разработчиками. Было также продемонстрировано использование инструментального средства *Unimod* для построения программ на основе автоматного подхода.

В заключение следует сказать, что на сегодняшний день все большее количество систем интегрируются в комплексы систем. Особенно это заметно в сфере электронного оборудования. Если же говорить о сигнализационных и противоугонных системах, то они все чаще устанавливаются прямо на заводе автопроизводителя, причем являются интегрированными в общую электронную систему автомобиля, наряду с мультимедийной системой, системой управления климатом в салоне и т.д. Учитывая это обстоятельство, можно утверждать, что автоматное программирование в целом и такое инструментальное средство, как *Unimod*, найдет применение в этой области и не только облегчит планирование, разработку и реализацию подобных систем, но и (при корректном использовании) исключит огромное количество вероятных ошибок, которое только увеличивается с ростом сложности систем.

ИСТОЧНИКИ

1. Сайт по автоматному программированию и мотивации к творчеству (<http://is.ifmo.ru/>)
2. Сайт проекта *Unimod* на *SourceForge* (<http://sourceforge.net/projects/unimod/>)
3.

Статья об автоматном программировании в Википедии (http://ru.wikipedia.org/wiki/Автоматное_программирование)
--
4. Статья о *Switch*-технологии в Википедии
(<http://ru.wikipedia.org/wiki/Switch-технология>)
5. Страница *Unimod*-проекта «Система управления автомобильной сигнализацией» на сайте по автоматному программированию
(<http://is.ifmo.ru/unimod-projects/alarmsystem/>)
6. Статья о противоугонных системах в Википедии
(http://ru.wikipedia.org/wiki/Противоугонная_система)

ПРИЛОЖЕНИЯ

Приложение 1. Описание Java-классов

В этом разделе будут приведены описания основных *Java*-классов и интерфейсов. Описание всех классов, сгенерированных инструментальным средством *Unimod*, здесь не приводится, поскольку их структура идентична.

1.1. *IRemoteGui.java*

```
package gui;

public interface IRemoteGui {

    void setMode(boolean isMenu);

    void setMenuTitle(int titleNum);

    void setMenuItem(int itemNum);

    void setLockState(boolean isLocked);

    void setSoundState(boolean isOn);

    void setShockState(boolean isShocked);

    void setDoorState(boolean isOpened);

}
```

1.2. *IRemoteControl.java*

```
package remote;

public interface IRemoteControl {

    void pressButton1();

    void pressButton2();

    void pressButton3();

}
```

1.3. *ICar.java*

```
package car;

public interface ICar {

    void setFrontDoorState(boolean isOpened);

    void setRearDoorState(boolean isOpened);

    void setHoodState(boolean isOpened);

    void setBootState(boolean isOpened);

    void setIgnitionState(boolean isOn);

    void setCentralLockState(boolean isLocked);

    /**
     * @param isLight
     * true if shock is light and false if strong.
     */
}
```

```

        */
        void setShock(boolean isLight);
    }

```

1.4. *MouseAreaWatcher.java*

```

package gui.panels;

import java.awt.Point;

import main.IReactable;

public class MouseAreaWatcher {
    private final Point myTopLeft;
    private final Point myBottomRight;
    private final IReactable myReactable;

    public MouseAreaWatcher(Point topLeft, Point bottomRight,
        IReactable reactable) {
        myTopLeft = topLeft;
        myBottomRight = bottomRight;
        myReactable = reactable;
    }

    public boolean isPointOver(Point p) {
        boolean activeX = ((p.getX() - myTopLeft.getX()) * (p.getX() -
myBottomRight
        .getX())) <= 0;
        boolean activeY = ((p.getY() - myTopLeft.getY()) * (p.getY() -
myBottomRight
        .getY())) <= 0;
        return (activeX && activeY);
    }

    public void react() {
        myReactable.react();
    }

    public boolean process(Point p) {
        if (isPointOver(p)) {
            myReactable.react();
            return true;
        }
        return false;
    }
}

```

1.5. *CarPanel.java*

```

package gui.panels;

import gui.states.MousePosState;

import javax.imageio.ImageIO;
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionListener;
import java.io.File;
import java.io.IOException;
import java.util.LinkedList;
import java.util.List;

```

```

public class CarPanel extends JPanel {
    private static final String MainBGImage = "main.png";

    private final Image myBackgroundImage;
    private final List<MousePosState> myMouseStates;
    private final MousePosState myBlinkingLampState;
    private Image myDoorPatchImage = ImageIO.read(new File("media/images",
        "doors.png"));

    private int myNumDoorsOpen;
    private BlinkerThread myBlinkerThread;

    public CarPanel(List<MousePosState> carStates, MousePosState
    blinkingLampState) throws IOException {
        myBackgroundImage = ImageIO.read(new File("media/images",
    MainBGImage));
        myMouseStates = new LinkedList<MousePosState>(carStates);
        myBlinkingLampState = blinkingLampState;
        myDoorPatchImage =
gui.states.Transparency.makeColorTransparent(myDoorPatchImage, new Color(200,
    20, 200));

        MousePosListener positionAdapter = new MousePosListener();
        addMouseListener(positionAdapter);
        addMouseMotionListener(positionAdapter);

        new Thread(myBlinkerThread).start();

        setPreferredSize(new Dimension(800, 536));
        setMinimumSize(getPreferredSize());
        setMaximumSize(getPreferredSize());
    }

    public void blink(int times) {
        if (myBlinkerThread != null) {
            myBlinkerThread.stop();
        }
        myBlinkerThread = new BlinkerThread(times);
        new Thread(myBlinkerThread).start();
    }

    public void startBlinking() {
        blink(99);
    }

    public void paint(Graphics g) {
        g.drawImage(myBackgroundImage, 0, 0, 800, 536, null);
        for (MousePosState state : myMouseStates) {
            if (state.isActivated())
                state.drawImage(g);
        }
        if (myNumDoorsOpen == 2) {
            getGraphics().drawImage(myDoorPatchImage, 491, 133,
                myDoorPatchImage.getWidth(null),
                myDoorPatchImage.getHeight(null), null);
        }
    }

    private class BlinkerThread implements Runnable {
        private int myCounter;

        public BlinkerThread(int times) {

```

```

        myCounter = times;
    }

    public void stop() {
        myCounter = 0;
    }

    public void run() {
        while (myCounter > 0) {
            myBlinkingLampState.react();
            myBlinkingLampState.animate(getGraphics());
            myBlinkingLampState.react();
            myBlinkingLampState.animate(getGraphics());
            myCounter--;
        }
    }
}

private class MousePosListener extends MouseAdapter implements
    MouseMotionListener {

    public void mouseClicked(MouseEvent e) {
        for (MousePosState state : myMouseStates) {
            if (state.isActiveOver(e.getPoint())) {
                state.react();
                if (state.isDoor() && state.isActivated())
                    myNumDoorsOpen++;
                if ((myNumDoorsOpen == 2) && (state.isDoor())) {
                    state.drawImage(getGraphics());
                } else {
                    Graphics g = getGraphics();
                    state.animate(g);
                }
                break;
            }
        }
        myNumDoorsOpen = 0;
        for (MousePosState state : myMouseStates) {
            if (state.isActivated()) {
                if (state.isDoor())
                    myNumDoorsOpen++;
                state.drawImage(getGraphics());
            }
        }
        if (myNumDoorsOpen == 2) {
            getGraphics().drawImage(myDoorPatchImage, 491, 133,
                myDoorPatchImage.getWidth(null),
                myDoorPatchImage.getHeight(null), null);
        }
    }

    public void mouseMoved(MouseEvent e) {

        setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
        for (MousePosState state : myMouseStates) {
            if (state.isActiveOver(e.getPoint())) {

                setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
            }
        }
    }

    public void mouseDragged(MouseEvent e) {

```

```

        // Just to be compatible with Java 1.5
    }
}

```

1.6. *RemoteControlPanel.java*

```

package gui.panels;

import gui.states.MousePosState;

import java.awt.Cursor;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionListener;
import java.io.File;
import java.io.IOException;
import java.util.LinkedList;
import java.util.List;

import javax.imageio.ImageIO;
import javax.swing.*;

public class RemoteControlPanel extends JPanel {
    private static final String MainBGImage = "remote.png";

    private final Image myBackgroundImage;
    private final List<MousePosState> myMouseStates;

    public RemoteControlPanel(List<MousePosState> remoteStates) throws
    IOException {
        myBackgroundImage = ImageIO.read(new File("media/images/remote",
MainBGImage));
        myMouseStates = new LinkedList<MousePosState>(remoteStates);

        setPreferredSize(new Dimension(361, 200));
        setMinimumSize(getPreferredSize());
        setMaximumSize(getPreferredSize());

        MousePosListener positionAdapter = new MousePosListener();
        addMouseListener(positionAdapter);
        addMouseMotionListener(positionAdapter);
    }

    public void paint(Graphics g) {
        g.drawImage(myBackgroundImage, 0, 0, 361, 200, null);
    }

    private class MousePosListener extends MouseAdapter implements
        MouseMotionListener {

        public void mouseClicked(MouseEvent e) {
            for (MousePosState state : myMouseStates) {
                if (state.isActiveOver(e.getPoint())) {
                    state.react();
                }
            }
        }

        public void mouseMoved(MouseEvent e) {

```

```

setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    for (MousePosState state : myMouseStates) {
        if (state.isActiveOver(e.getPoint())) {

setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        }
    }
}

public void mouseDragged(MouseEvent e) {
    // Just to be compatible with Java 1.5
}
}
}

```

1.7. RemoteControlBackPanel.java

```

package gui.panels;

import gui.IRemoteGui;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.JPanel;

import main.CarGui;

import gui.states.Transparency;

public class RemoteControlBackPanel extends JPanel implements IRemoteGui {
    private static final String PATH = "media/images/remote/";
    private static final String MainBGImage = "remoteBack.png";
    private static final String navigationImage = "navigation.png";

    private static final String exitMode = "exitMode.png";

    private static final String mainMenu = "mainMenu.png";
    private static final String silentMenu = "silentMenu.png";
    private static final String sensorMenu = "sensorMenu.png";

    private static final String silentMode = "silentMode.png";
    private static final String sensorMode = "sensorMode.png";

    private static final String enableMode = "enableMode.png";
    private static final String disableMode = "disableMode.png";

    private static final String lowMode = "lowMode.png";
    private static final String middleMode = "middleMode.png";
    private static final String highMode = "highMode.png";

    private static final String lockON = "lockON.png";
    private static final String lockOFF = "lockOFF.png";
    private static final String soundON = "soundON.png";
    private static final String soundOFF = "soundOFF.png";
    private static final String kickON = "kickON.png";
    private static final String kickOFF = "kickOFF.png";
    private static final String doorsON = "doorsON.png";

```



```

private static final String doorsOFF = "doorsOFF.png";

private final Image myBackgroundImage = ImageIO.read(new File(PATH,
    MainBGImage));
private final Image myNavigationImage =
Transparency.makeColorTransparent(
    ImageIO.read(new File(PATH, navigationImage)), new
Color(200, 20,
        200));

private final Image mainMenuImage = ImageIO.read(new File(PATH,
mainMenu));
private final Image silentMenuImage = ImageIO.read(new File(PATH,
    silentMenu));
private final Image sensorMenuImage = ImageIO.read(new File(PATH,
    sensorMenu));
private final Image[] myTitles = new Image[] { mainMenuImage,
    silentMenuImage, sensorMenuImage };

private final Image silentModeImage = ImageIO.read(new File(PATH
    + "/mainMenu", silentMode));
private final Image sensorModeImage = ImageIO.read(new File(PATH
    + "/mainMenu", sensorMode));
private final Image enableModeImage = ImageIO.read(new File(PATH
    + "/silentMenu", enableMode));
private final Image disableModeImage = ImageIO.read(new File(PATH
    + "/silentMenu", disableMode));
private final Image lowModeImage = ImageIO.read(new File(PATH
    + "/sensorMenu", lowMode));
private final Image middleModeImage = ImageIO.read(new File(PATH
    + "/sensorMenu", middleMode));
private final Image highModeImage = ImageIO.read(new File(PATH
    + "/sensorMenu", highMode));
private final Image exitModeImage = ImageIO.read(new File(PATH,
exitMode));

private final Image[] myItems = new Image[] { silentModeImage,
    sensorModeImage, enableModeImage, disableModeImage,
lowModeImage,
    middleModeImage, highModeImage, exitModeImage };

private final Image lockONImage = ImageIO.read(new File(PATH + "/picts",
    lockON));
private final Image lockOFFImage = ImageIO.read(new File(PATH +
"/picts",
    lockOFF));
private final Image soundONImage = ImageIO.read(new File(PATH +
"/picts",
    soundON));
private final Image soundOFFImage = ImageIO.read(new File(PATH +
"/picts",
    soundOFF));
private final Image kickONImage = ImageIO.read(new File(PATH + "/picts",
    kickON));
private final Image kickOFFImage = ImageIO.read(new File(PATH +
"/picts",
    kickOFF));
private final Image doorsONImage = ImageIO.read(new File(PATH +
"/picts",
    doorsON));
private final Image doorsOFFImage = ImageIO.read(new File(PATH +
"/picts",
    doorsOFF));

```

```

private boolean myLockIsLocked = false;
private boolean mySoundIsOn = true;
private boolean myShockIsOn = false;
private boolean myDoorIsOpened = true;
private boolean myMenuState = false;
private int myCurrentTitle = 0;
private int myCurrentItem = 0;

public RemoteControlBackPanel() throws IOException {
    setPreferredSize(new Dimension(361, 200));
    setMinimumSize(getPreferredSize());
    setMaximumSize(getPreferredSize());

    CarGui.setRemote(this);
}

public void paint(Graphics g) {
    if (myMenuState) {
        g.drawImage(myNavigationImage, 0, 0, 361, 200, null);
        paintTitle(g);
        paintItem(g);
    } else {
        g.drawImage(myBackgroundImage, 0, 0, 361, 200, null);
        paintIcons(g);
    }
}

private void paintTitle(Graphics g) {
    g
        .drawImage(myTitles[myCurrentTitle], 68, 27,
            myTitles[myCurrentTitle].getWidth(null),
            myTitles[myCurrentTitle].getHeight(null),
null);
}

private void paintItem(Graphics g) {
    g.drawImage(myItems[myCurrentItem], 71, 58, myItems[myCurrentItem]
        .getWidth(null),
myItems[myCurrentItem].getHeight(null), null);
}

private void paintIcons(Graphics g) {
    if (myLockIsLocked) {
        g.drawImage(lockONImage, 96, 62, lockONImage.getWidth(null),
            lockONImage.getHeight(null), null);
    } else {
        g.drawImage(lockOFFImage, 96, 62,
lockOFFImage.getWidth(null),
            lockOFFImage.getHeight(null), null);
    }
    if (mySoundIsOn) {
        g.drawImage(soundONImage, 35, 58,
soundONImage.getWidth(null),
            soundONImage.getHeight(null), null);
    } else {
        g.drawImage(soundOFFImage, 35, 58,
soundOFFImage.getWidth(null),
            soundOFFImage.getHeight(null), null);
    }
    if (myShockIsOn) {
        g.drawImage(kickONImage, 154, 43,
kickONImage.getWidth(null),

```

```

        kickONImage.getHeight(null), null);
    } else {
        g.drawImage(kickOFFImage, 154, 43,
kickOFFImage.getWidth(null),
        kickOFFImage.getHeight(null), null);
    }
    if (myDoorIsOpened) {
        g.drawImage(doorsONImage, 200, 30,
doorsONImage.getWidth(null),
        doorsONImage.getHeight(null), null);
    } else {
        g.drawImage(doorsOFFImage, 200, 30,
doorsOFFImage.getWidth(null),
        doorsOFFImage.getHeight(null), null);
    }
}

public void setMode(boolean isMenu) {
    myMenuState = isMenu;
    repaint();
}

public void setMenuItem(int itemNum) {
    myCurrentItem = itemNum;
    repaint();
}

public void setMenuTitle(int titleNum) {
    myCurrentTitle = titleNum;
    repaint();
}

public void setLockState(boolean isLocked) {
    myLockIsLocked = isLocked;
    repaint();
}

public void setSoundState(boolean isOn) {
    mySoundIsOn = isOn;
    repaint();
}

public void setShockState(boolean isShocked) {
    myShockIsOn = isShocked;
    repaint();
}

public void setDoorState(boolean isOpened) {
    myDoorIsOpened = isOpened;
    repaint();
}
}

```

1.8. Siren.java

```

package siren;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import javazoom.jl.decoder.JavaLayerException;
import javazoom.jl.player.Player;

```

```

public class Siren {

    private final static String BEEP1 = "media/sounds/oneBeep.mp3";
    private final static String BEEP2 = "media/sounds/twoBeeps.mp3";
    private final static String BEEP3 = "media/sounds/threeBeeps.mp3";
    private final static String BEEP4 = "media/sounds/fourBeeps.mp3";
    private final static String ALARM = "media/sounds/alarm.mp3";
    private final static String DING = "media/sounds/ding.mp3";

    private static boolean lastWasAlarm = false;

    private static PlayerThread myPlayerThread;

    public static void ding() {
        if (lastWasAlarm && myPlayerThread.isAlive()) {
            return;
        }
        new PlayerThread(DING, 1).start();
    }

    public static void alarm(int amount) {
        if (lastWasAlarm && myPlayerThread.isAlive()) {
            myPlayerThread.incCounter(amount);
            return;
        }
        lastWasAlarm = true;
        if (myPlayerThread != null) {
            myPlayerThread.stopPlaying();
        }
        myPlayerThread = new PlayerThread(ALARM, amount);
        myPlayerThread.start();
    }

    public static void beep(int times) {
        if ((times < 1) || (times > 4)) {
            return;
        }
        if (myPlayerThread != null) {
            myPlayerThread.stopPlaying();
        }
        switch (times) {
            case 1:
                myPlayerThread = new PlayerThread(BEEP1, 1);
                break;
            case 2:
                myPlayerThread = new PlayerThread(BEEP2, 1);
                break;
            case 3:
                myPlayerThread = new PlayerThread(BEEP3, 1);
                break;
            case 4:
                myPlayerThread = new PlayerThread(BEEP4, 1);
                break;
        }

        new Thread(myPlayerThread).start();
    }

    private static class PlayerThread extends Thread {

        private Player myPlayer;
        private InputStream myInput;
    }
}

```

```

private final String myFileName;
private int myCounter;

public PlayerThread(String fileName, int amount) {
    myFileName = fileName;
    myCounter = amount;
}

public void incCounter(int amount) {
    if (amount > myCounter) {
        myCounter = amount;
    }
}

public void run() {
    while (myCounter > 0) {
        resetInput();
        try {
            myPlayer.play();
            myCounter--;
        } catch (JavaLayerException e) {
            System.out.println("Error while playing " + e);
            return;
        }
    }
}

private void resetInput() {
    try {
        myInput = new FileInputStream(myFileName);
        myPlayer = new Player(myInput);
    } catch (IOException e) {
        System.out.println("File not found " + e);
    } catch (JavaLayerException e) {
        System.out.println("Unable to play file " + e);
    }
}

public void stopPlaying() {
    myCounter = 0;
    myPlayer.close();
}
}
}

```

Приложение 2. XML-описание модели

В этом разделе приводится описание XML-модели сигнализационной системы, то есть описание, собственно, автоматов, управляющих работой всех компонентов системы.

2.1. A1Car.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><!DOCTYPE model PUBLIC
"-//eVelopers Corp.//DTD State machine model V1.0//EN"
"http://www.evelopers.com/dtd/unimod/statemachine.dtd">
<model name="Modell">
    <controlledObject class="car.CarCO" name="o1"/>
    <controlledObject class="car.SignalsCO" name="o2"/>
    <controlledObject class="car.CarBlockCO" name="o4"/>
    <controlledObject class="remote.RemoteCO" name="o3"/>
    <eventProvider class="car.CarStateEP" name="p1">
        <association clientRole="p1" targetRef="A1Car"/>
    </eventProvider>

```

```

<eventProvider class="remote.RemoteEP" name="p2">
  <association clientRole="p2" targetRef="A1Car"/>
</eventProvider>
<eventProvider class="remote.RemotePhysicalEP" name="p3">
  <association clientRole="p3" targetRef="A1Car"/>
</eventProvider>
<eventProvider class="car.CarBlockEP" name="p4">
  <association clientRole="p4" targetRef="A1Car"/>
</eventProvider>
<rootStateMachine>
  <stateMachineRef name="A1Car"/>
</rootStateMachine>
<stateMachine name="A1Car">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association clientRole="A1Car" supplierRole="o1" targetRef="o1"/>
  <association clientRole="A1Car" supplierRole="A2System"
targetRef="A2System"/>
  <state name="Top" type="NORMAL">
    <state name="s2" type="INITIAL"/>
    <state name="s1" type="NORMAL">
      <stateMachineRef name="A2System"/>
      <stateMachineRef name="A3Remote"/>
    </state>
  </state>
  <transition sourceRef="s2" targetRef="s1"/>
  <transition event="e101" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z1"/>
  </transition>
  <transition event="e102" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z2"/>
  </transition>
  <transition event="e103" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z3"/>
  </transition>
  <transition event="e104" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z4"/>
  </transition>
  <transition event="e105" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z5"/>
  </transition>
  <transition event="e111" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z11"/>
  </transition>
  <transition event="e110" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z10"/>
  </transition>
  <transition event="e109" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z9"/>
  </transition>
  <transition event="e106" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z6"/>
  </transition>
  <transition event="e107" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z7"/>
  </transition>
  <transition event="e108" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z8"/>
  </transition>
  <transition event="e112" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z12"/>
  </transition>
</stateMachine>

```

```

<stateMachine name="A2System">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association clientRole="A2System" supplierRole="o1" targetRef="o1"/>
  <association clientRole="A2System" supplierRole="o2" targetRef="o2"/>
  <association clientRole="A2System" supplierRole="o4" targetRef="o4"/>
  <association clientRole="A2System" supplierRole="A3Remote"
targetRef="A3Remote"/>
  <state name="Top" type="NORMAL">
    <state name="In drive" type="NORMAL"/>
    <state name="In drive / manual locks" type="NORMAL"/>
    <state name="Ready" type="NORMAL">
      <outputAction ident="o2.z10"/>
    </state>
    <state name="s1" type="INITIAL"/>
    <state name="Shock sensor activated" type="NORMAL">
      <outputAction ident="o2.z13"/>
    </state>
    <state name="Guard normal" type="NORMAL">
      <outputAction ident="o2.z9"/>
    </state>
    <state name="Panic" type="NORMAL">
      <outputAction ident="o2.z8"/>
      <outputAction ident="o2.z11"/>
    </state>
    <state name="Alarm activated" type="NORMAL">
      <outputAction ident="o2.z7"/>
      <outputAction ident="o2.z12"/>
    </state>
  </state>
  <transition event="e202" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z2"/>
  </transition>
  <transition event="e201" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z1"/>
  </transition>
  <transition event="e111" sourceRef="In drive" targetRef="In drive /
manual locks"/>
  <transition event="e112" sourceRef="In drive" targetRef="In drive /
manual locks"/>
  <transition event="e110" sourceRef="In drive" targetRef="Ready">
    <outputAction ident="o4.z2"/>
  </transition>
  <transition event="e201" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
    <outputAction ident="o4.z1"/>
  </transition>
  <transition event="e202" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
    <outputAction ident="o4.z2"/>
  </transition>
  <transition event="e110" sourceRef="In drive / manual locks"
targetRef="Ready"/>
  <transition event="e109" sourceRef="Ready" targetRef="In drive">
    <outputAction ident="o4.z1"/>
  </transition>
  <transition event="e202" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z2"/>
    <outputAction ident="o2.z2"/>
  </transition>
  <transition event="e206" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z7"/>
    <outputAction ident="o2.z4"/>
  </transition>

```

```

</transition>
<transition event="e204" sourceRef="Ready" targetRef="Ready">
  <outputAction ident="o4.z5"/>
  <outputAction ident="o2.z4"/>
</transition>
<transition event="e205" sourceRef="Ready" targetRef="Ready">
  <outputAction ident="o4.z6"/>
  <outputAction ident="o2.z4"/>
</transition>
<transition event="e201" guard="!o1.x1" sourceRef="Ready"
targetRef="Guard normal">
  <outputAction ident="o4.z1"/>
  <outputAction ident="o4.z3"/>
  <outputAction ident="o2.z3"/>
</transition>
<transition event="e201" guard="o1.x1" sourceRef="Ready" targetRef="Guard
normal">
  <outputAction ident="o4.z1"/>
  <outputAction ident="o4.z3"/>
  <outputAction ident="o2.z1"/>
</transition>
<transition sourceRef="s1" targetRef="Ready"/>
<transition event="e202" sourceRef="Shock sensor activated"
targetRef="Ready">
  <outputAction ident="o4.z2"/>
  <outputAction ident="o4.z4"/>
  <outputAction ident="o2.z3"/>
</transition>
<transition event="e113" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
  <outputAction ident="o2.z3"/>
</transition>
<transition event="e114" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
  <outputAction ident="o2.z6"/>
</transition>
<transition event="e201" sourceRef="Shock sensor activated"
targetRef="Guard normal">
  <outputAction ident="o2.z1"/>
</transition>
<transition event="e203" sourceRef="Shock sensor activated"
targetRef="Panic"/>
<transition event="e101" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
<transition event="e103" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
<transition event="e105" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
<transition event="e107" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
<transition event="e109" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
<transition event="e111" sourceRef="Shock sensor activated"
targetRef="Alarm activated">
  <outputAction ident="o4.z1"/>
</transition>
<transition event="e202" sourceRef="Guard normal" targetRef="Ready">
  <outputAction ident="o4.z2"/>
  <outputAction ident="o4.z4"/>
  <outputAction ident="o2.z2"/>
</transition>
<transition event="e114" guard="o4.x1 &gt; 1" sourceRef="Guard normal"
targetRef="Shock sensor activated">

```



```

        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e113" guard="o4.x1 > 2" sourceRef="Guard normal"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e201" sourceRef="Guard normal" targetRef="Guard
normal">
        <outputAction ident="o2.z5"/>
    </transition>
    <transition event="e113" guard="o4.x1 == 2" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e114" guard="o4.x1 <= 1" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e113" guard="o4.x1 < 2" sourceRef="Guard normal"
targetRef="Guard normal"/>
    <transition event="e203" sourceRef="Guard normal" targetRef="Panic"/>
    <transition event="e111" sourceRef="Guard normal" targetRef="Alarm
activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e109" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e105" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e107" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e103" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e101" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e201" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e202" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e111" sourceRef="Panic" targetRef="Panic">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="Alarm activated" targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z4"/>
    </transition>
    <transition event="e201" sourceRef="Alarm activated" targetRef="Guard
normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e203" sourceRef="Alarm activated" targetRef="Panic"/>
    <transition event="e101" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e103" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e105" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e107" sourceRef="Alarm activated" targetRef="Alarm
activated"/>

```

```

    <transition event="e109" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e111" sourceRef="Alarm activated" targetRef="Alarm
activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e113" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e114" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
</stateMachine>
<stateMachine name="A3Remote">
    <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
    <association clientRole="A3Remote" supplierRole="o3" targetRef="o3"/>
    <state name="Top" type="NORMAL">
        <state name="Silent mode on" type="NORMAL">
            <outputAction ident="o3.z16"/>
        </state>
        <state name="Silent mode off" type="NORMAL">
            <outputAction ident="o3.z17"/>
        </state>
        <state name="Exit silent mode" type="NORMAL">
            <outputAction ident="o3.z21"/>
        </state>
        <state name="Shock 1" type="NORMAL">
            <outputAction ident="o3.z18"/>
        </state>
        <state name="Shock 2" type="NORMAL">
            <outputAction ident="o3.z19"/>
        </state>
        <state name="Shock 3" type="NORMAL">
            <outputAction ident="o3.z20"/>
        </state>
        <state name="Exit shock" type="NORMAL">
            <outputAction ident="o3.z21"/>
        </state>
        <state name="Menu silent mode" type="NORMAL">
            <outputAction ident="o3.z14"/>
        </state>
        <state name="Menu shock" type="NORMAL">
            <outputAction ident="o3.z15"/>
        </state>
        <state name="Exit menu" type="NORMAL">
            <outputAction ident="o3.z21"/>
        </state>
        <state name="s2" type="INITIAL"/>
        <state name="Normal" type="NORMAL">
            <outputAction ident="o3.z27"/>
            <outputAction ident="o3.z25"/>
            <outputAction ident="o3.z9"/>
        </state>
        <state name="Alarm" type="NORMAL">
            <outputAction ident="o3.z9"/>
            <outputAction ident="o3.z28"/>
        </state>
        <state name="Guard" type="NORMAL">
            <outputAction ident="o3.z10"/>
            <outputAction ident="o3.z25"/>
            <outputAction ident="o3.z26"/>
        </state>
        <state name="Shocked" type="NORMAL">
            <outputAction ident="o3.z24"/>

```

```

        <outputAction ident="o3.z28"/>
    </state>
    <state name="Panic" type="NORMAL"/>
</state>
<transition event="e303" sourceRef="Silent mode on" targetRef="Silent
mode on">
    <outputAction ident="o3.z28"/>
    <outputAction ident="o3.z4"/>
</transition>
<transition event="e301" sourceRef="Silent mode on" targetRef="Silent
mode off"/>
<transition event="e302" sourceRef="Silent mode on" targetRef="Exit
silent mode"/>
<transition event="e302" sourceRef="Silent mode off" targetRef="Silent
mode on"/>
<transition event="e303" sourceRef="Silent mode off" targetRef="Silent
mode off">
    <outputAction ident="o3.z5"/>
    <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Silent mode off" targetRef="Exit
silent mode"/>
<transition event="e301" sourceRef="Exit silent mode" targetRef="Silent
mode on"/>
<transition event="e302" sourceRef="Exit silent mode" targetRef="Silent
mode off"/>
<transition event="e303" sourceRef="Exit silent mode" targetRef="Menu
silent mode">
    <outputAction ident="o3.z11"/>
</transition>
<transition event="e303" sourceRef="Shock 1" targetRef="Shock 1">
    <outputAction ident="o3.z6"/>
    <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Shock 1" targetRef="Shock 2"/>
<transition event="e302" sourceRef="Shock 1" targetRef="Exit shock"/>
<transition event="e302" sourceRef="Shock 2" targetRef="Shock 1"/>
<transition event="e303" sourceRef="Shock 2" targetRef="Shock 2">
    <outputAction ident="o3.z7"/>
    <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Shock 2" targetRef="Shock 3"/>
<transition event="e302" sourceRef="Shock 3" targetRef="Shock 2"/>
<transition event="e303" sourceRef="Shock 3" targetRef="Shock 3">
    <outputAction ident="o3.z8"/>
    <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Shock 3" targetRef="Exit shock"/>
<transition event="e301" sourceRef="Exit shock" targetRef="Shock 1"/>
<transition event="e302" sourceRef="Exit shock" targetRef="Shock 3"/>
<transition event="e303" sourceRef="Exit shock" targetRef="Menu shock">
    <outputAction ident="o3.z11"/>
</transition>
<transition event="e303" guard="!o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode on">
    <outputAction ident="o3.z12"/>
</transition>
<transition event="e303" guard="o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode off">
    <outputAction ident="o3.z12"/>
</transition>
<transition event="e301" sourceRef="Menu silent mode" targetRef="Menu
shock"/>

```

```

    <transition event="e302" sourceRef="Menu silent mode" targetRef="Exit
menu"/>
    <transition event="e303" sourceRef="Menu shock" targetRef="Shock 1">
        <outputAction ident="o3.z13"/>
    </transition>
    <transition event="e302" sourceRef="Menu shock" targetRef="Menu silent
mode"/>
    <transition event="e301" sourceRef="Menu shock" targetRef="Exit menu"/>
    <transition event="e301" sourceRef="Exit menu" targetRef="Menu silent
mode"/>
    <transition event="e302" sourceRef="Exit menu" targetRef="Menu shock"/>
    <transition event="e303" sourceRef="Exit menu" targetRef="Normal">
        <outputAction ident="o3.z23"/>
    </transition>
    <transition sourceRef="s2" targetRef="Normal"/>
    <transition event="e303" sourceRef="Normal" targetRef="Menu silent mode">
        <outputAction ident="o3.z22"/>
        <outputAction ident="o3.z11"/>
    </transition>
    <transition event="e301" sourceRef="Normal" targetRef="Normal">
        <outputAction ident="o3.z1"/>
    </transition>
    <transition event="e302" sourceRef="Normal" targetRef="Normal">
        <outputAction ident="o3.z2"/>
    </transition>
    <transition event="e401" sourceRef="Normal" targetRef="Guard"/>
    <transition event="e402" sourceRef="Alarm" targetRef="Normal"/>
    <transition event="e301" sourceRef="Alarm" targetRef="Alarm">
        <outputAction ident="o3.z1"/>
    </transition>
    <transition event="e302" sourceRef="Alarm" targetRef="Alarm">
        <outputAction ident="o3.z2"/>
    </transition>
    <transition event="e303" sourceRef="Alarm" targetRef="Alarm">
        <outputAction ident="o3.z3"/>
    </transition>
    <transition event="e401" sourceRef="Alarm" targetRef="Guard"/>
    <transition event="e403" sourceRef="Alarm" targetRef="Panic"/>
    <transition event="e402" sourceRef="Guard" targetRef="Normal"/>
    <transition event="e404" sourceRef="Guard" targetRef="Alarm"/>
    <transition event="e302" sourceRef="Guard" targetRef="Guard">
        <outputAction ident="o3.z2"/>
    </transition>
    <transition event="e303" sourceRef="Guard" targetRef="Guard">
        <outputAction ident="o3.z3"/>
    </transition>
    <transition event="e301" sourceRef="Guard" targetRef="Guard">
        <outputAction ident="o3.z1"/>
    </transition>
    <transition event="e405" sourceRef="Guard" targetRef="Shocked"/>
    <transition event="e403" sourceRef="Guard" targetRef="Panic"/>
    <transition event="e402" sourceRef="Shocked" targetRef="Normal"/>
    <transition event="e401" sourceRef="Shocked" targetRef="Guard"/>
    <transition event="e301" sourceRef="Shocked" targetRef="Shocked">
        <outputAction ident="o3.z1"/>
    </transition>
    <transition event="e302" sourceRef="Shocked" targetRef="Shocked">
        <outputAction ident="o3.z2"/>
    </transition>
    <transition event="e303" sourceRef="Shocked" targetRef="Shocked">
        <outputAction ident="o3.z3"/>
    </transition>
    <transition event="e403" sourceRef="Shocked" targetRef="Panic"/>

```

```

    <transition event="e401" sourceRef="Panic" targetRef="Guard"/>
    <transition event="e301" sourceRef="Panic" targetRef="Panic">
      <outputAction ident="o3.z1"/>
    </transition>
    <transition event="e302" sourceRef="Panic" targetRef="Panic">
      <outputAction ident="o3.z2"/>
    </transition>
  </stateMachine>
</model>

```

2.2. *A2System.xml*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><!DOCTYPE model PUBLIC
"-//evelopers Corp.//DTD State machine model V1.0//EN"
"http://www.evelopers.com/dtd/unimod/statemachine.dtd">
<model name="Modell">
  <controlledObject class="car.CarCO" name="o1"/>
  <controlledObject class="car.SignalsCO" name="o2"/>
  <controlledObject class="car.CarBlockCO" name="o4"/>
  <controlledObject class="remote.RemoteCO" name="o3"/>
  <eventProvider class="car.CarStateEP" name="p1">
    <association clientRole="p1" targetRef="A1Car"/>
  </eventProvider>
  <eventProvider class="remote.RemoteEP" name="p2">
    <association clientRole="p2" targetRef="A1Car"/>
  </eventProvider>
  <eventProvider class="remote.RemotePhysicalEP" name="p3">
    <association clientRole="p3" targetRef="A1Car"/>
  </eventProvider>
  <eventProvider class="car.CarBlockEP" name="p4">
    <association clientRole="p4" targetRef="A1Car"/>
  </eventProvider>
  <rootStateMachine>
    <stateMachineRef name="A2System"/>
  </rootStateMachine>
  <stateMachine name="A1Car">
    <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
    <association clientRole="A1Car" supplierRole="o1" targetRef="o1"/>
    <association clientRole="A1Car" supplierRole="A2System"
targetRef="A2System"/>
    <state name="Top" type="NORMAL">
      <state name="s2" type="INITIAL"/>
      <state name="s1" type="NORMAL">
        <stateMachineRef name="A2System"/>
        <stateMachineRef name="A3Remote"/>
      </state>
    </state>
    <transition sourceRef="s2" targetRef="s1"/>
    <transition event="e101" sourceRef="s1" targetRef="s1">
      <outputAction ident="o1.z1"/>
    </transition>
    <transition event="e102" sourceRef="s1" targetRef="s1">
      <outputAction ident="o1.z2"/>
    </transition>
    <transition event="e103" sourceRef="s1" targetRef="s1">
      <outputAction ident="o1.z3"/>
    </transition>
    <transition event="e104" sourceRef="s1" targetRef="s1">
      <outputAction ident="o1.z4"/>
    </transition>
    <transition event="e105" sourceRef="s1" targetRef="s1">
      <outputAction ident="o1.z5"/>
    </transition>

```

```

<transition event="e111" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z11"/>
</transition>
<transition event="e110" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z10"/>
</transition>
<transition event="e109" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z9"/>
</transition>
<transition event="e106" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z6"/>
</transition>
<transition event="e107" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z7"/>
</transition>
<transition event="e108" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z8"/>
</transition>
<transition event="e112" sourceRef="s1" targetRef="s1">
  <outputAction ident="o1.z12"/>
</transition>
</stateMachine>
<stateMachine name="A2System">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association clientRole="A2System" supplierRole="o1" targetRef="o1"/>
  <association clientRole="A2System" supplierRole="o2" targetRef="o2"/>
  <association clientRole="A2System" supplierRole="o4" targetRef="o4"/>
  <association clientRole="A2System" supplierRole="A3Remote"
targetRef="A3Remote"/>
  <state name="Top" type="NORMAL">
    <state name="In drive" type="NORMAL"/>
    <state name="In drive / manual locks" type="NORMAL"/>
    <state name="Ready" type="NORMAL">
      <outputAction ident="o2.z10"/>
    </state>
    <state name="s1" type="INITIAL"/>
    <state name="Shock sensor activated" type="NORMAL">
      <outputAction ident="o2.z13"/>
    </state>
    <state name="Guard normal" type="NORMAL">
      <outputAction ident="o2.z9"/>
    </state>
    <state name="Panic" type="NORMAL">
      <outputAction ident="o2.z8"/>
      <outputAction ident="o2.z11"/>
    </state>
    <state name="Alarm activated" type="NORMAL">
      <outputAction ident="o2.z7"/>
      <outputAction ident="o2.z12"/>
    </state>
  </state>
  <transition event="e202" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z2"/>
  </transition>
  <transition event="e201" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z1"/>
  </transition>
  <transition event="e111" sourceRef="In drive" targetRef="In drive /
manual locks"/>
  <transition event="e112" sourceRef="In drive" targetRef="In drive /
manual locks"/>
  <transition event="e110" sourceRef="In drive" targetRef="Ready">

```

```

        <outputAction ident="o4.z2"/>
    </transition>
    <transition event="e201" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
        <outputAction ident="o4.z2"/>
    </transition>
    <transition event="e110" sourceRef="In drive / manual locks"
targetRef="Ready"/>
    <transition event="e109" sourceRef="Ready" targetRef="In drive">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="Ready" targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o2.z2"/>
    </transition>
    <transition event="e206" sourceRef="Ready" targetRef="Ready">
        <outputAction ident="o4.z7"/>
        <outputAction ident="o2.z4"/>
    </transition>
    <transition event="e204" sourceRef="Ready" targetRef="Ready">
        <outputAction ident="o4.z5"/>
        <outputAction ident="o2.z4"/>
    </transition>
    <transition event="e205" sourceRef="Ready" targetRef="Ready">
        <outputAction ident="o4.z6"/>
        <outputAction ident="o2.z4"/>
    </transition>
    <transition event="e201" guard="!o1.x1" sourceRef="Ready"
targetRef="Guard normal">
        <outputAction ident="o4.z1"/>
        <outputAction ident="o4.z3"/>
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e201" guard="o1.x1" sourceRef="Ready" targetRef="Guard
normal">
        <outputAction ident="o4.z1"/>
        <outputAction ident="o4.z3"/>
        <outputAction ident="o2.z1"/>
    </transition>
    <transition sourceRef="s1" targetRef="Ready"/>
    <transition event="e202" sourceRef="Shock sensor activated"
targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e113" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e114" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e201" sourceRef="Shock sensor activated"
targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>

```

```

    <transition event="e203" sourceRef="Shock sensor activated"
targetRef="Panic"/>
    <transition event="e101" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e103" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e105" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e107" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e109" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e111" sourceRef="Shock sensor activated"
targetRef="Alarm activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="Guard normal" targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z2"/>
    </transition>
    <transition event="e114" guard="o4.x1 > 1" sourceRef="Guard normal"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e113" guard="o4.x1 > 2" sourceRef="Guard normal"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e201" sourceRef="Guard normal" targetRef="Guard
normal">
        <outputAction ident="o2.z5"/>
    </transition>
    <transition event="e113" guard="o4.x1 == 2" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e114" guard="o4.x1 <= 1" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e113" guard="o4.x1 < 2" sourceRef="Guard normal"
targetRef="Guard normal"/>
    <transition event="e203" sourceRef="Guard normal" targetRef="Panic"/>
    <transition event="e111" sourceRef="Guard normal" targetRef="Alarm
activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e109" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e105" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e107" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e103" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e101" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e201" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e202" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>

```



```

</transition>
<transition event="e111" sourceRef="Panic" targetRef="Panic">
  <outputAction ident="o4.z1"/>
</transition>
<transition event="e202" sourceRef="Alarm activated" targetRef="Ready">
  <outputAction ident="o4.z2"/>
  <outputAction ident="o4.z4"/>
  <outputAction ident="o2.z4"/>
</transition>
<transition event="e201" sourceRef="Alarm activated" targetRef="Guard
normal">
  <outputAction ident="o2.z1"/>
</transition>
<transition event="e203" sourceRef="Alarm activated" targetRef="Panic"/>
<transition event="e101" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e103" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e105" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e107" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e109" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e111" sourceRef="Alarm activated" targetRef="Alarm
activated">
  <outputAction ident="o4.z1"/>
</transition>
<transition event="e113" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
<transition event="e114" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
</stateMachine>
<stateMachine name="A3Remote">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association clientRole="A3Remote" supplierRole="o3" targetRef="o3"/>
  <state name="Top" type="NORMAL">
    <state name="Silent mode on" type="NORMAL">
      <outputAction ident="o3.z16"/>
    </state>
    <state name="Silent mode off" type="NORMAL">
      <outputAction ident="o3.z17"/>
    </state>
    <state name="Exit silent mode" type="NORMAL">
      <outputAction ident="o3.z21"/>
    </state>
    <state name="Shock 1" type="NORMAL">
      <outputAction ident="o3.z18"/>
    </state>
    <state name="Shock 2" type="NORMAL">
      <outputAction ident="o3.z19"/>
    </state>
    <state name="Shock 3" type="NORMAL">
      <outputAction ident="o3.z20"/>
    </state>
    <state name="Exit shock" type="NORMAL">
      <outputAction ident="o3.z21"/>
    </state>
    <state name="Menu silent mode" type="NORMAL">
      <outputAction ident="o3.z14"/>
    </state>
    <state name="Menu shock" type="NORMAL">

```

```

    <outputAction ident="o3.z15"/>
  </state>
  <state name="Exit menu" type="NORMAL">
    <outputAction ident="o3.z21"/>
  </state>
  <state name="s2" type="INITIAL"/>
  <state name="Normal" type="NORMAL">
    <outputAction ident="o3.z27"/>
    <outputAction ident="o3.z25"/>
    <outputAction ident="o3.z9"/>
  </state>
  <state name="Alarm" type="NORMAL">
    <outputAction ident="o3.z9"/>
    <outputAction ident="o3.z28"/>
  </state>
  <state name="Guard" type="NORMAL">
    <outputAction ident="o3.z10"/>
    <outputAction ident="o3.z25"/>
    <outputAction ident="o3.z26"/>
  </state>
  <state name="Shocked" type="NORMAL">
    <outputAction ident="o3.z24"/>
    <outputAction ident="o3.z28"/>
  </state>
  <state name="Panic" type="NORMAL"/>
</state>
<transition event="e303" sourceRef="Silent mode on" targetRef="Silent
mode on">
  <outputAction ident="o3.z28"/>
  <outputAction ident="o3.z4"/>
</transition>
<transition event="e301" sourceRef="Silent mode on" targetRef="Silent
mode off"/>
<transition event="e302" sourceRef="Silent mode on" targetRef="Exit
silent mode"/>
<transition event="e302" sourceRef="Silent mode off" targetRef="Silent
mode on"/>
<transition event="e303" sourceRef="Silent mode off" targetRef="Silent
mode off">
  <outputAction ident="o3.z5"/>
  <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Silent mode off" targetRef="Exit
silent mode"/>
<transition event="e301" sourceRef="Exit silent mode" targetRef="Silent
mode on"/>
<transition event="e302" sourceRef="Exit silent mode" targetRef="Silent
mode off"/>
<transition event="e303" sourceRef="Exit silent mode" targetRef="Menu
silent mode">
  <outputAction ident="o3.z11"/>
</transition>
<transition event="e303" sourceRef="Shock 1" targetRef="Shock 1">
  <outputAction ident="o3.z6"/>
  <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Shock 1" targetRef="Shock 2"/>
<transition event="e302" sourceRef="Shock 1" targetRef="Exit shock"/>
<transition event="e302" sourceRef="Shock 2" targetRef="Shock 1"/>
<transition event="e303" sourceRef="Shock 2" targetRef="Shock 2">
  <outputAction ident="o3.z7"/>
  <outputAction ident="o3.z28"/>
</transition>

```

```

<transition event="e301" sourceRef="Shock 2" targetRef="Shock 3"/>
<transition event="e302" sourceRef="Shock 3" targetRef="Shock 2"/>
<transition event="e303" sourceRef="Shock 3" targetRef="Shock 3">
  <outputAction ident="o3.z8"/>
  <outputAction ident="o3.z28"/>
</transition>
<transition event="e301" sourceRef="Shock 3" targetRef="Exit shock"/>
<transition event="e301" sourceRef="Exit shock" targetRef="Shock 1"/>
<transition event="e302" sourceRef="Exit shock" targetRef="Shock 3"/>
<transition event="e303" sourceRef="Exit shock" targetRef="Menu shock">
  <outputAction ident="o3.z11"/>
</transition>
<transition event="e303" guard="!o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode on">
  <outputAction ident="o3.z12"/>
</transition>
<transition event="e303" guard="o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode off">
  <outputAction ident="o3.z12"/>
</transition>
<transition event="e301" sourceRef="Menu silent mode" targetRef="Menu
shock"/>
<transition event="e302" sourceRef="Menu silent mode" targetRef="Exit
menu"/>
<transition event="e303" sourceRef="Menu shock" targetRef="Shock 1">
  <outputAction ident="o3.z13"/>
</transition>
<transition event="e302" sourceRef="Menu shock" targetRef="Menu silent
mode"/>
<transition event="e301" sourceRef="Menu shock" targetRef="Exit menu"/>
<transition event="e301" sourceRef="Exit menu" targetRef="Menu silent
mode"/>
<transition event="e302" sourceRef="Exit menu" targetRef="Menu shock"/>
<transition event="e303" sourceRef="Exit menu" targetRef="Normal">
  <outputAction ident="o3.z23"/>
</transition>
<transition sourceRef="s2" targetRef="Normal"/>
<transition event="e303" sourceRef="Normal" targetRef="Menu silent mode">
  <outputAction ident="o3.z22"/>
  <outputAction ident="o3.z11"/>
</transition>
<transition event="e301" sourceRef="Normal" targetRef="Normal">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Normal" targetRef="Normal">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e401" sourceRef="Normal" targetRef="Guard"/>
<transition event="e402" sourceRef="Alarm" targetRef="Normal"/>
<transition event="e301" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z3"/>
</transition>
<transition event="e401" sourceRef="Alarm" targetRef="Guard"/>
<transition event="e403" sourceRef="Alarm" targetRef="Panic"/>
<transition event="e402" sourceRef="Guard" targetRef="Normal"/>
<transition event="e404" sourceRef="Guard" targetRef="Alarm"/>
<transition event="e302" sourceRef="Guard" targetRef="Guard">

```

```

    <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Guard" targetRef="Guard">
    <outputAction ident="o3.z3"/>
</transition>
<transition event="e301" sourceRef="Guard" targetRef="Guard">
    <outputAction ident="o3.z1"/>
</transition>
<transition event="e405" sourceRef="Guard" targetRef="Shocked"/>
<transition event="e403" sourceRef="Guard" targetRef="Panic"/>
<transition event="e402" sourceRef="Shocked" targetRef="Normal"/>
<transition event="e401" sourceRef="Shocked" targetRef="Guard"/>
<transition event="e301" sourceRef="Shocked" targetRef="Shocked">
    <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Shocked" targetRef="Shocked">
    <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Shocked" targetRef="Shocked">
    <outputAction ident="o3.z3"/>
</transition>
<transition event="e403" sourceRef="Shocked" targetRef="Panic"/>
<transition event="e401" sourceRef="Panic" targetRef="Guard"/>
<transition event="e301" sourceRef="Panic" targetRef="Panic">
    <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Panic" targetRef="Panic">
    <outputAction ident="o3.z2"/>
</transition>
</stateMachine>
</model>

```

2.3. *A3Remote.xml*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><!DOCTYPE model PUBLIC
"-//evelopers Corp.//DTD State machine model V1.0//EN"
"http://www.evelopers.com/dtd/unimod/statemachine.dtd">
<model name="Modell">
    <controlledObject class="car.CarCO" name="o1"/>
    <controlledObject class="car.SignalsCO" name="o2"/>
    <controlledObject class="car.CarBlockCO" name="o4"/>
    <controlledObject class="remote.RemoteCO" name="o3"/>
    <eventProvider class="car.CarStateEP" name="p1">
        <association clientRole="p1" targetRef="A1Car"/>
    </eventProvider>
    <eventProvider class="remote.RemoteEP" name="p2">
        <association clientRole="p2" targetRef="A1Car"/>
    </eventProvider>
    <eventProvider class="remote.RemotePhysicalEP" name="p3">
        <association clientRole="p3" targetRef="A1Car"/>
    </eventProvider>
    <eventProvider class="car.CarBlockEP" name="p4">
        <association clientRole="p4" targetRef="A1Car"/>
    </eventProvider>
    <rootStateMachine>
        <stateMachineRef name="A3Remote"/>
    </rootStateMachine>
    <stateMachine name="A1Car">
        <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
        <association clientRole="A1Car" supplierRole="o1" targetRef="o1"/>
        <association clientRole="A1Car" supplierRole="A2System"
targetRef="A2System"/>
        <state name="Top" type="NORMAL">

```

```

    <state name="s2" type="INITIAL"/>
    <state name="s1" type="NORMAL">
      <stateMachineRef name="A2System"/>
      <stateMachineRef name="A3Remote"/>
    </state>
  </state>
  <transition sourceRef="s2" targetRef="s1"/>
  <transition event="e101" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z1"/>
  </transition>
  <transition event="e102" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z2"/>
  </transition>
  <transition event="e103" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z3"/>
  </transition>
  <transition event="e104" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z4"/>
  </transition>
  <transition event="e105" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z5"/>
  </transition>
  <transition event="e111" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z11"/>
  </transition>
  <transition event="e110" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z10"/>
  </transition>
  <transition event="e109" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z9"/>
  </transition>
  <transition event="e106" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z6"/>
  </transition>
  <transition event="e107" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z7"/>
  </transition>
  <transition event="e108" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z8"/>
  </transition>
  <transition event="e112" sourceRef="s1" targetRef="s1">
    <outputAction ident="o1.z12"/>
  </transition>
</stateMachine>
<stateMachine name="A2System">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association clientRole="A2System" supplierRole="o1" targetRef="o1"/>
  <association clientRole="A2System" supplierRole="o2" targetRef="o2"/>
  <association clientRole="A2System" supplierRole="o4" targetRef="o4"/>
  <association clientRole="A2System" supplierRole="A3Remote"
targetRef="A3Remote"/>
  <state name="Top" type="NORMAL">
    <state name="In drive" type="NORMAL"/>
    <state name="In drive / manual locks" type="NORMAL"/>
    <state name="Ready" type="NORMAL">
      <outputAction ident="o2.z10"/>
    </state>
  </state>
  <state name="s1" type="INITIAL"/>
  <state name="Shock sensor activated" type="NORMAL">
    <outputAction ident="o2.z13"/>
  </state>
  <state name="Guard normal" type="NORMAL">

```

```

        <outputAction ident="o2.z9"/>
    </state>
    <state name="Panic" type="NORMAL">
        <outputAction ident="o2.z8"/>
        <outputAction ident="o2.z11"/>
    </state>
    <state name="Alarm activated" type="NORMAL">
        <outputAction ident="o2.z7"/>
        <outputAction ident="o2.z12"/>
    </state>
</state>
<transition event="e202" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z2"/>
</transition>
<transition event="e201" sourceRef="In drive" targetRef="In drive">
    <outputAction ident="o4.z1"/>
</transition>
<transition event="e111" sourceRef="In drive" targetRef="In drive /
manual locks"/>
<transition event="e112" sourceRef="In drive" targetRef="In drive /
manual locks"/>
<transition event="e110" sourceRef="In drive" targetRef="Ready">
    <outputAction ident="o4.z2"/>
</transition>
<transition event="e201" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
    <outputAction ident="o4.z1"/>
</transition>
<transition event="e202" sourceRef="In drive / manual locks"
targetRef="In drive / manual locks">
    <outputAction ident="o4.z2"/>
</transition>
<transition event="e110" sourceRef="In drive / manual locks"
targetRef="Ready"/>
<transition event="e109" sourceRef="Ready" targetRef="In drive">
    <outputAction ident="o4.z1"/>
</transition>
<transition event="e202" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z2"/>
    <outputAction ident="o2.z2"/>
</transition>
<transition event="e206" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z7"/>
    <outputAction ident="o2.z4"/>
</transition>
<transition event="e204" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z5"/>
    <outputAction ident="o2.z4"/>
</transition>
<transition event="e205" sourceRef="Ready" targetRef="Ready">
    <outputAction ident="o4.z6"/>
    <outputAction ident="o2.z4"/>
</transition>
<transition event="e201" guard="!o1.x1" sourceRef="Ready"
targetRef="Guard normal">
    <outputAction ident="o4.z1"/>
    <outputAction ident="o4.z3"/>
    <outputAction ident="o2.z3"/>
</transition>
<transition event="e201" guard="o1.x1" sourceRef="Ready" targetRef="Guard
normal">
    <outputAction ident="o4.z1"/>
    <outputAction ident="o4.z3"/>

```

```

        <outputAction ident="o2.z1"/>
    </transition>
    <transition sourceRef="s1" targetRef="Ready"/>
    <transition event="e202" sourceRef="Shock sensor activated"
targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e113" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e114" sourceRef="Shock sensor activated"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e201" sourceRef="Shock sensor activated"
targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e203" sourceRef="Shock sensor activated"
targetRef="Panic"/>
    <transition event="e101" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e103" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e105" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e107" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e109" sourceRef="Shock sensor activated"
targetRef="Alarm activated"/>
    <transition event="e111" sourceRef="Shock sensor activated"
targetRef="Alarm activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="Guard normal" targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z2"/>
    </transition>
    <transition event="e114" guard="o4.x1 > 1" sourceRef="Guard normal"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e113" guard="o4.x1 > 2" sourceRef="Guard normal"
targetRef="Shock sensor activated">
        <outputAction ident="o2.z6"/>
    </transition>
    <transition event="e201" sourceRef="Guard normal" targetRef="Guard
normal">
        <outputAction ident="o2.z5"/>
    </transition>
    <transition event="e113" guard="o4.x1 == 2" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>
    <transition event="e114" guard="o4.x1 <= 1" sourceRef="Guard normal"
targetRef="Guard normal">
        <outputAction ident="o2.z3"/>
    </transition>

```

```

    <transition event="e113" guard="o4.x1 &lt; 2" sourceRef="Guard normal"
targetRef="Guard normal"/>
    <transition event="e203" sourceRef="Guard normal" targetRef="Panic"/>
    <transition event="e111" sourceRef="Guard normal" targetRef="Alarm
activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e109" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e105" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e107" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e103" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e101" sourceRef="Guard normal" targetRef="Alarm
activated"/>
    <transition event="e201" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e202" sourceRef="Panic" targetRef="Guard normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e111" sourceRef="Panic" targetRef="Panic">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e202" sourceRef="Alarm activated" targetRef="Ready">
        <outputAction ident="o4.z2"/>
        <outputAction ident="o4.z4"/>
        <outputAction ident="o2.z4"/>
    </transition>
    <transition event="e201" sourceRef="Alarm activated" targetRef="Guard
normal">
        <outputAction ident="o2.z1"/>
    </transition>
    <transition event="e203" sourceRef="Alarm activated" targetRef="Panic"/>
    <transition event="e101" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e103" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e105" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e107" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e109" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e111" sourceRef="Alarm activated" targetRef="Alarm
activated">
        <outputAction ident="o4.z1"/>
    </transition>
    <transition event="e113" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
    <transition event="e114" sourceRef="Alarm activated" targetRef="Alarm
activated"/>
</stateMachine>
<stateMachine name="A3Remote">
    <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
    <association clientRole="A3Remote" supplierRole="o3" targetRef="o3"/>
    <state name="Top" type="NORMAL">
        <state name="Silent mode on" type="NORMAL">
            <outputAction ident="o3.z16"/>
        </state>
    </state>

```



```

<state name="Silent mode off" type="NORMAL">
  <outputAction ident="o3.z17"/>
</state>
<state name="Exit silent mode" type="NORMAL">
  <outputAction ident="o3.z21"/>
</state>
<state name="Shock 1" type="NORMAL">
  <outputAction ident="o3.z18"/>
</state>
<state name="Shock 2" type="NORMAL">
  <outputAction ident="o3.z19"/>
</state>
<state name="Shock 3" type="NORMAL">
  <outputAction ident="o3.z20"/>
</state>
<state name="Exit shock" type="NORMAL">
  <outputAction ident="o3.z21"/>
</state>
<state name="Menu silent mode" type="NORMAL">
  <outputAction ident="o3.z14"/>
</state>
<state name="Menu shock" type="NORMAL">
  <outputAction ident="o3.z15"/>
</state>
<state name="Exit menu" type="NORMAL">
  <outputAction ident="o3.z21"/>
</state>
<state name="s2" type="INITIAL"/>
<state name="Normal" type="NORMAL">
  <outputAction ident="o3.z27"/>
  <outputAction ident="o3.z25"/>
  <outputAction ident="o3.z9"/>
</state>
<state name="Alarm" type="NORMAL">
  <outputAction ident="o3.z9"/>
  <outputAction ident="o3.z28"/>
</state>
<state name="Guard" type="NORMAL">
  <outputAction ident="o3.z10"/>
  <outputAction ident="o3.z25"/>
  <outputAction ident="o3.z26"/>
</state>
<state name="Shocked" type="NORMAL">
  <outputAction ident="o3.z24"/>
  <outputAction ident="o3.z28"/>
</state>
<state name="Panic" type="NORMAL"/>
</state>
<transition event="e303" sourceRef="Silent mode on" targetRef="Silent
mode on">
  <outputAction ident="o3.z28"/>
  <outputAction ident="o3.z4"/>
</transition>
<transition event="e301" sourceRef="Silent mode on" targetRef="Silent
mode off"/>
<transition event="e302" sourceRef="Silent mode on" targetRef="Exit
silent mode"/>
<transition event="e302" sourceRef="Silent mode off" targetRef="Silent
mode on"/>
<transition event="e303" sourceRef="Silent mode off" targetRef="Silent
mode off">
  <outputAction ident="o3.z5"/>
  <outputAction ident="o3.z28"/>

```

```

    </transition>
    <transition event="e301" sourceRef="Silent mode off" targetRef="Exit
silent mode"/>
    <transition event="e301" sourceRef="Exit silent mode" targetRef="Silent
mode on"/>
    <transition event="e302" sourceRef="Exit silent mode" targetRef="Silent
mode off"/>
    <transition event="e303" sourceRef="Exit silent mode" targetRef="Menu
silent mode">
        <outputAction ident="o3.z11"/>
    </transition>
    <transition event="e303" sourceRef="Shock 1" targetRef="Shock 1">
        <outputAction ident="o3.z6"/>
        <outputAction ident="o3.z28"/>
    </transition>
    <transition event="e301" sourceRef="Shock 1" targetRef="Shock 2"/>
    <transition event="e302" sourceRef="Shock 1" targetRef="Exit shock"/>
    <transition event="e302" sourceRef="Shock 2" targetRef="Shock 1"/>
    <transition event="e303" sourceRef="Shock 2" targetRef="Shock 2">
        <outputAction ident="o3.z7"/>
        <outputAction ident="o3.z28"/>
    </transition>
    <transition event="e301" sourceRef="Shock 2" targetRef="Shock 3"/>
    <transition event="e302" sourceRef="Shock 3" targetRef="Shock 2"/>
    <transition event="e303" sourceRef="Shock 3" targetRef="Shock 3">
        <outputAction ident="o3.z8"/>
        <outputAction ident="o3.z28"/>
    </transition>
    <transition event="e301" sourceRef="Shock 3" targetRef="Exit shock"/>
    <transition event="e301" sourceRef="Exit shock" targetRef="Shock 1"/>
    <transition event="e302" sourceRef="Exit shock" targetRef="Shock 3"/>
    <transition event="e303" sourceRef="Exit shock" targetRef="Menu shock">
        <outputAction ident="o3.z11"/>
    </transition>
    <transition event="e303" guard="!o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode on">
        <outputAction ident="o3.z12"/>
    </transition>
    <transition event="e303" guard="o3.x1" sourceRef="Menu silent mode"
targetRef="Silent mode off">
        <outputAction ident="o3.z12"/>
    </transition>
    <transition event="e301" sourceRef="Menu silent mode" targetRef="Menu
shock"/>
    <transition event="e302" sourceRef="Menu silent mode" targetRef="Exit
menu"/>
    <transition event="e303" sourceRef="Menu shock" targetRef="Shock 1">
        <outputAction ident="o3.z13"/>
    </transition>
    <transition event="e302" sourceRef="Menu shock" targetRef="Menu silent
mode"/>
    <transition event="e301" sourceRef="Menu shock" targetRef="Exit menu"/>
    <transition event="e301" sourceRef="Exit menu" targetRef="Menu silent
mode"/>
    <transition event="e302" sourceRef="Exit menu" targetRef="Menu shock"/>
    <transition event="e303" sourceRef="Exit menu" targetRef="Normal">
        <outputAction ident="o3.z23"/>
    </transition>
    <transition sourceRef="s2" targetRef="Normal"/>
    <transition event="e303" sourceRef="Normal" targetRef="Menu silent mode">
        <outputAction ident="o3.z22"/>
        <outputAction ident="o3.z11"/>
    </transition>

```

```

<transition event="e301" sourceRef="Normal" targetRef="Normal">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Normal" targetRef="Normal">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e401" sourceRef="Normal" targetRef="Guard"/>
<transition event="e402" sourceRef="Alarm" targetRef="Normal"/>
<transition event="e301" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Alarm" targetRef="Alarm">
  <outputAction ident="o3.z3"/>
</transition>
<transition event="e401" sourceRef="Alarm" targetRef="Guard"/>
<transition event="e403" sourceRef="Alarm" targetRef="Panic"/>
<transition event="e402" sourceRef="Guard" targetRef="Normal"/>
<transition event="e404" sourceRef="Guard" targetRef="Alarm"/>
<transition event="e302" sourceRef="Guard" targetRef="Guard">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Guard" targetRef="Guard">
  <outputAction ident="o3.z3"/>
</transition>
<transition event="e301" sourceRef="Guard" targetRef="Guard">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e405" sourceRef="Guard" targetRef="Shocked"/>
<transition event="e403" sourceRef="Guard" targetRef="Panic"/>
<transition event="e402" sourceRef="Shocked" targetRef="Normal"/>
<transition event="e401" sourceRef="Shocked" targetRef="Guard"/>
<transition event="e301" sourceRef="Shocked" targetRef="Shocked">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Shocked" targetRef="Shocked">
  <outputAction ident="o3.z2"/>
</transition>
<transition event="e303" sourceRef="Shocked" targetRef="Shocked">
  <outputAction ident="o3.z3"/>
</transition>
<transition event="e403" sourceRef="Shocked" targetRef="Panic"/>
<transition event="e401" sourceRef="Panic" targetRef="Guard"/>
<transition event="e301" sourceRef="Panic" targetRef="Panic">
  <outputAction ident="o3.z1"/>
</transition>
<transition event="e302" sourceRef="Panic" targetRef="Panic">
  <outputAction ident="o3.z2"/>
</transition>
</stateMachine>
</model>

```