Projektübung Informatik

Universelle Infrarot-Fernsteuerung

3. Objekt Orientierte Analyse

Zweite, überarbeitete Version

Gruppe 13: R. Zingg
M. Kaufmann
M. Stämpfli

Klasse: E3d

Datum: 16.5.1997

Inhaltsverzeichnis

3 Objektorientierte Analyse

3.1 Begriffserklärungen und Definitionen

Aktuelles Objekt:
Das aktuelle Objekt in UIF ist entweder eine Fernsteuerung (SRC) oder eine Sequenz. Anhand der Farbgebung des Titelbalkens des Fensters kann man erkennen, welches Objekt aktuell ist. Es kann jedoch immer nur maximal ein aktuelles Objekt geben.

Markiertes Objekt:
In einer Sequenz kann dies ein markiertes Textfragment sein, welches durch Inversion der Farben markiert ist. In einer Fernsteuerung (SRC) kann es ein markiertes RCButton, RCLine oder RCText Objekt sein, welches durch einen Rahmen markiert wird. Es kann jedoch immer nur maximal ein markiertes Objekt geben. Das markierte Objekt befindet sich zudem immer in einem aktuellen Objekt.

IR-Code
Sequenz von Infrarotblitzen, die ein Zielgerät veranlasst, eine bestimmte Funktion auszuführen.

Zielgerät
Allgemein ein elektronisches Gerät welches über Infrarot gesteuert werden kann. Dies könnte z.B. eine HiFi-Anlage, ein Video oder ein TV sein.

Grundpulslänge:
Grösster gemeinsamer Teiler der Pulslängen in ms, welche im entsprechenden IR-Code vorkommen.

Nutzcode:
Der Nutzcode ist der IR-Code, welcher die Information enthält, welche Taste gedrückt wurde. Dieser Code wird als erstes gesendet, wenn eine Taste gedrückt wird. Nach dem Nutzcode wird, je nach Typ (Sony oder Kenwood), der Nutzcode oder der Folgecode wiederholt gesendet, bis die Taste wieder losgelassen wird.

Folgecode:
Der Folgecode ist der IR-Code, welcher bei Fernsteuerungen (HRC) vom Typ 'Kenwood' benutzt wird, um anzuzeigen, dass eine Taste gedrückt wird. Der Code selbst enthält keine Information. Damit das Zielgerät erkennt, welche Taste gedrückt wird, muss zuvor der Nutzcode empfangen worden sein.

3.2 Klassenspezifikationen

3.2.1 Die Klasse MainObject

Von der abstrakten Klasse MainObject werden die Klassen CommandSequence und RemoteControl abgeleitet.

Attribute:

dirty:	Gibt an, ob seit dem erstellen oder letzten mal speichern Änderungen vorgenommen wurden.
path:	Enthält den Pfad der Datei, in welcher dieses Objekt gespeichert wurde.
name:	Enthält den Dateinamen der Datei, in welcher dieses Objekt gespeichert wurde, oder den von UIF zugewiesenen temporären Namen.
nameTemp:	Gibt an, ob der Name in 'name' temporär ist, also von UIF gegeben, oder vom Anwender angegeben wurde. Dies hat z.B. dann einen Einfluss, wenn das Objekt gespeichert werden soll. Ist der Name temporär, so muss ein Dateiname angegeben werden. Ist er nicht temporär, so wird der in den Attributen path und name gespeicherten Pfad und Dateiname verwendet.

Methoden:

Load():	Ladet ein auf einem Laufwerk als Datei gespeichertes Objekt in den Speicher.
Save():	Speichert das Objekt in einer Datei auf einem Laufwerk.
Show():	Zeichnet das Objekt auf den Bildschirm.

3.2.2 Die Klasse RemoteControl

Ein Objekt der Klasse RemoteControl repräsentiert die eigentliche Fernsteuerung auf dem Bildschirm. Sie beinhaltet beliebig viele RCElement Objekte.
Die Klasse RemoteControl erbt von der abstrakten Klasse MainObject.

Assoziation:

aktRCElement:	Das momentan markierte RCElement Objekt in der Fernsteuerung.

Methoden:

AddRCButton():	Fügt der Fernsteuerung ein neues RCButton Objekt zu und zeigt es an.
AddRCLine():	Fügt der Fernsteuerung ein neues RCLine Objekt zu und zeigt es an.
AddRCText():	Fügt der Fernsteuerung ein neues RCText Objekt zu und zeigt es an.

3.2.3 Die Klasse RCElement

Von der abstrakten Klasse RCElement werden die Klassen RCButton, RCText und RCLine abgeleitet.
Ein Objekt der Klasse RCElement gehört immer zu einem Objekt der Klasse RemoteControl.

Attribut:

pos:	Die Position des Elementes relativ zur linken oberen Ecke der Fernsteuerung.

Methoden:

ReadAndCreate():	Erzeugt eine neues RCElement und hold die Attribute aus einer Datei.
Write():	Schreibt die Attribute des Objektes in eine Datei.
Show():	Zeichnet das Objekt auf den Bildschirm.

3.2.4 Die Klasse RCLine

Ein Objekt der Klasse RCLine erscheint auf dem Bildschirm als eine Linie auf der Fernsteuerung (SRC), die zur Verbesserung der Übersichtlichkeit der Fernsteuerung dient.
Ein Objekt der Klasse RCLine gehört immer zu einem Objekt der Klasse RemoteControl.
Die Klasse RCLine erbt von der abstrakten Klasse RCElement

Attribut:

size:	Gibt den Abstand des Endpunktes zum Anfangspunkt der Linie an.

3.2.5 Die Klasse RCText

Ein Objekt der Klasse RCText erscheint auf dem Bildschirm als ein Text auf der Fernsteuerung (SRC), der zur Beschriftung der Fernsteuerung dient um die Übersichtlichkeit zu verbessern.
Ein Objekt der Klasse RCText gehört immer zu einem Objekt der Klasse RemoteControl.
Die Klasse RCText erbt von der abstrakten Klasse RCElement.

Attribut:

label:	Enthält den Text, welcher auf die Fernsteuerung geschrieben werden soll.

3.2.6 Die Klasse RCButton

Ein Objekt der Klasse RCButton erscheint auf der Fernsteuerung (SRC) als Taste. Jeder RCButton beinhaltet ein Objekt der Klasse IRInfo welches alle Informationen beinhaltet die nötig sind, um einen IR-Code zu generieren. Wird der RCButton betätigt, wird ISES veranlasst, den entsprechenden IR-Code zu senden und die dem entsprechenden RCButton eigene Funktion beim Zielgerät auszulösen. Dazu wird das eigene Objekt IRInfo benötigt. Ein Objekt der Klasse RCButton gehört immer zu einem Objekt der Klasse RemoteControl. Die Klasse RCButton erbt von der abstrakten Klasse RCElement.

Attribute:

key:	Gibt den internen Namen der Taste an. Dieser muss pro Fernsteuerung einmalig vorhanden sein (max. 15 Zeichen, keine Leerzeichen.). Dieser Wert wird benötigt, um die Taste von einer Sequenz aus anzusprechen.
label:	Text für die Beschriftung der Taste.
picture:	Pfad einer Bitmap - Datei (.bmp), welche auf der Taste angezeigt werden soll.

Has - Beziehung:

info:	Jeder RCButton besitzt ein IRInfo, in welchem alle Informationen gespeichert sind, um einen IR-Code zu generieren.

Methoden:

Press():	Der der Taste zugewiesene IR-Code wird von ISES ausgesendet.
Release():	Sendet den Befehl an ISES, die Übertragung abzubrechen.

3.2.7 Die Klasse IRInfo

Ein Objekt der Klasse IRInfo ist ein Teil eines Objektes der Klasse RCButton. Es enthält sämtliche Informationen über den IR-Code des zugehörigen RCButton. Es kann ISES veranlassen, den IR-Code auszusenden oder einen IR-Code von einer Fernsteuerung (HRC) einzuscannen.
Ein Objekt der Klasse IRInfo kennt immer ein Objekt der Klasse ISES.

Attribute:

typ:	Spezifiziert die Art der Übertragung. (Sony / Kenwood) Def.: S
modFreq:	Modulationsfrequenz. Gibt die Frequenz an, mit welcher die Daten moduliert werden. (max. Modulationsfrequenz = 42 kHz) Def.: 40 kHz
timeBase:	Gibt die Grundpulslänge des verwendeten Codes an. Def.: 0.6 ms
repeat:	Anzahl Wiederholungen (0-254, 255 = endlos) von - Code1 beim Sony - typ - Code2 beim Kenwood - typ
code1:	Binärer Code (Nutzcode) in ASCII ("10011100.....") des - 1. Pulspaketes (Kenwood) - der Pulspakete (Sony)
code2:	Binärer Code (Folgecode) in ASCII ("10101010111") - nur beim Kenwood - typ erforderlich.

Kennen - Beziehung:

ISES:	Schnittstelle zur Hardware ISES.

Methoden:

Send ():	Sendet ‘seinen’ IR-Code über ISES an das Zielgerät.
ScanAndGet ():	Setzt die Attribute entsprechend der von ISES von einer Fernsteuerung (HRC) eingelesenen Daten.
AbortSend():	Bricht das Senden eines IR-Codes durch ISES ab.
ReadAndCreate():	Erzeugt eine neue IRInfo und hold die Attribute aus einer Datei.
Write():	Schreibt die Attribute des Objektes in eine Datei.

3.2.8 Die Klasse ISES

Ein Objekt der Klasse ISES ist die Schnittstelle zwischen der Hardware ISES und der Software UIF. Damit ist sie sozusagen der Treiber für unsere Hardware ISES.
Zudem hat ISES die Funktionalität, die von ISES (HW) erhaltenen Daten auszuwerten und in ein IRInfo zu wandeln.
Es können mehrere Objekte der Klasse ISES vorhanden sein und jedes Objekt der Klasse ISES kann von mehreren Objekten der Klasse IRInfo benützt werden.

Methoden:

SendIRCode():	Wandelt IRInfo in einen für ISES verständlichen Code um, und weist ISES an, ihn zu senden.
AbortSend():	Sendet ISES ein Signal (ASCII 'Q') um alle Aktivitäten abzubrechen.
ScanIRCode():	Weist ISES an, den IR-Code von einer Fernsteuerung (HRC) einzulesen und in ISES zu speichern.
GetIRCode():	Holt den in ISES gespeicherten IR-Code ab und verarbeitet ihn zu einer IRInfo.

3.2.9 Die Klasse CommandSequence

Ein Objekt der Klasse CommandSequence besteht aus beliebig vielen Objekten der Klasse Command. Am Bildschirm erscheint CommandSequence als ein Textfenster. Die Befehlssequenz kann aus den Befehlen PressButton, Wait und WaitUntil besteht. Die einzelnen Befehle sind in der Sequenz als Objekte der Klasse Command enthalten. Die Abarbeitung der Befehlssequenz kann vom Anwender oder vom Timer gestartet werden. Jede Sequenz hat genau einen Timer.

Attribut:

busy:	Wahr, wenn die Sequenz momentan ausgeführt wird. Das Objekt Sequenz ist für den Benützer gesperrt.

Has - Beziehung:

timer:	Der Timer der für das zeitversetzte Starten der Sequenz und zum pausieren der Sequenz dient.

Assoziation:

nextCommand:	Der nächste Befehl der abgearbeitet werden soll.

Methoden:

Execute():	Führt die Befehle von Anfang an aus.
Continue():	Führt die Befehle von der Position 'nextCommand' her aus.
Test():	Prüft alle 'Command' Objekte auf Syntax und überprüft ob alle verwendeten RemoteControl und RCButton Objekte vorhanden sind.
Stop():	Stoppt die Ausführung der Sequenz und stoppt den Timer, falls er läuft.
AddCommand():	Fügt der Sequenz ein neues Objekt der Klasse Command zu.

3.2.10 Die Klasse Command

Ein Objekt der Klasse Command ist ein Befehl in einer Sequenz.
Beliebig viele Objekte der Klasse Command gehören zu einem Objekt der Klasse CommandSequence, d.h. eine Sequenz kann aus beliebig vielen Kommandos bestehen.
Falls es ein 'PressButton' Befehl ist, kennt Command das verwendete Objekt der Klasse RCButton.

Attribute:

text:	Der Text des auszuführenden Befehls.

Methoden:

Test():	Testet den Befehl auf Syntax und überprüft beim Befehl 'PressButton', ob die angegebene Fernsteuerung und der angegebene Button vorhanden sind.
Execute():	Führt den Befehl in 'text' aus.
Show():	Zeigt den Befehl auf dem Bildschirm an.
ReadAndCreate():	Erzeugt ein neues Objekt der Klasse Command anhand der Informationen die aus einer Datei gelesen werden.
Write():	Speichert die Informationen des Objektes in einer Datei.

3.2.11 Die Klasse Timer

Ein Objekt der Klasse Timer ruft zur angegebenen Zeit die in command angegebene Methode auf. Ein Timer gehört zu einer CommandSequence.

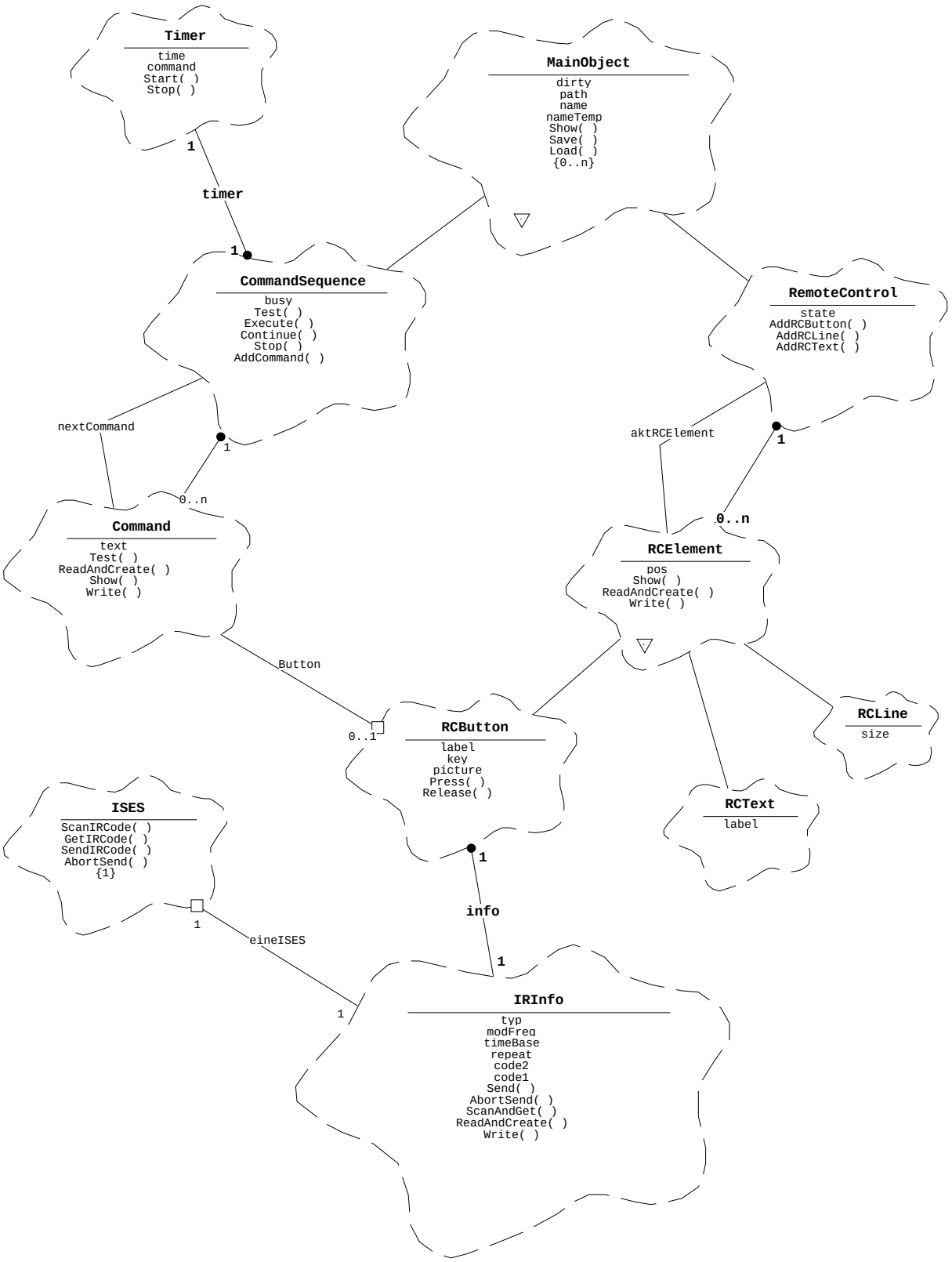
Attribute:

command:	Die Methode, die nach Ablauf der Zeit aufgerufen werden soll.
time:	Die Zeit, zu welcher der Timer die in command angegebene Methode aufrufen soll.

Methoden:

Start():	Setzt und startet den Timer.
Stop()	Stoppt den Timer.

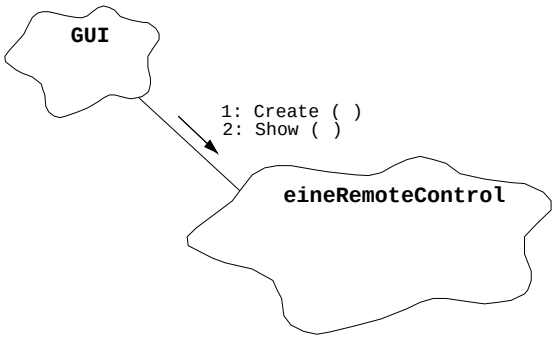
3.3 Das Klassendiagramm von UIF



3.4 Objektszenarien

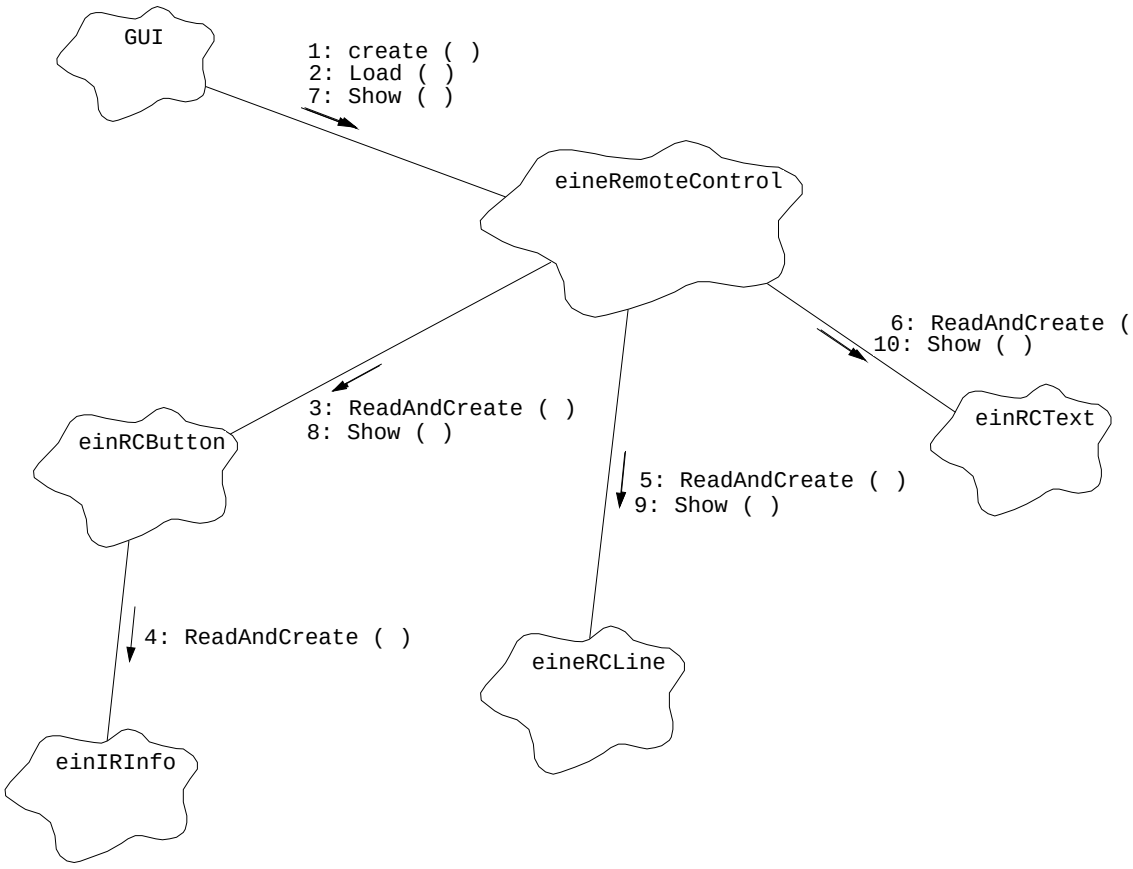
3.4.1 Objektszenario Fernsteuerung erstellen

Generiert eine neue, leere Fernbedienung und zeigt sie am Bildschirm an.



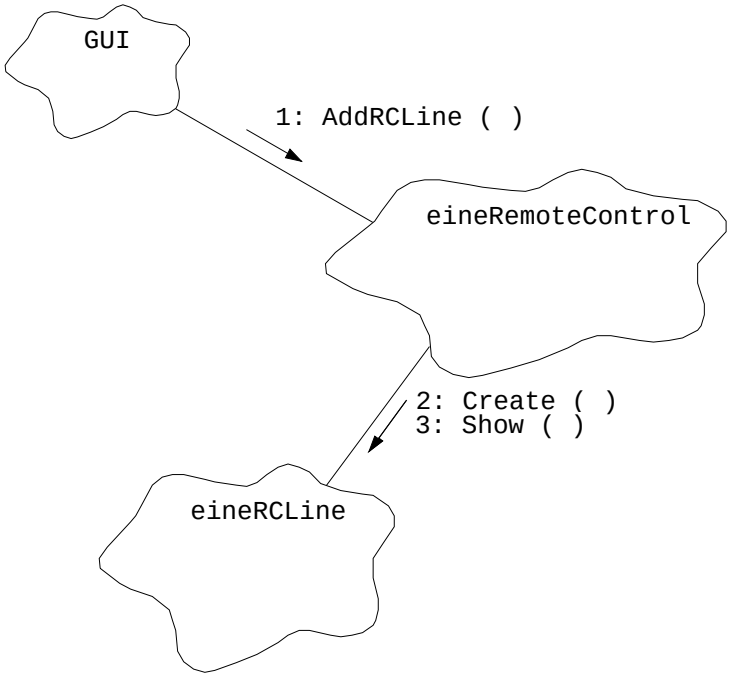
3.4.2 Objektszenario Fernsteuerung laden

Eine auf einem Laufwerk als Datei gespeicherte Fernsteuerung (SRC) mit einem Button, einer Linie und einem Text wird geladen und angezeigt.



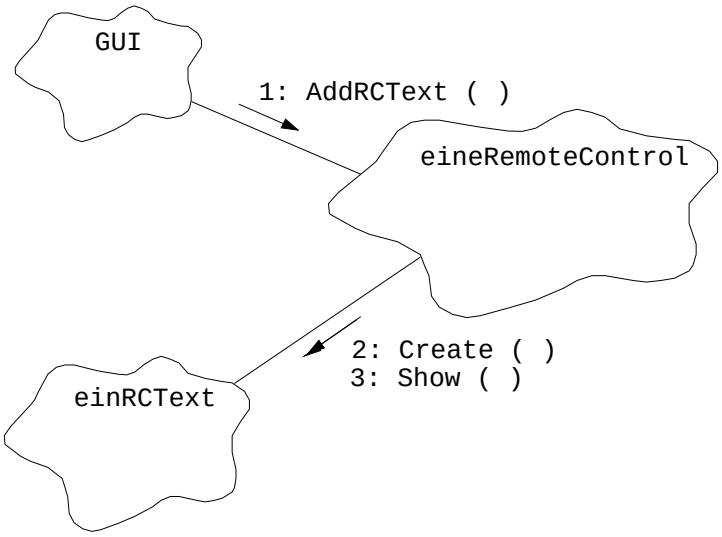
3.4.3 Objektszenario Linie erstellen

In der aktuellen Fernsteuerung (SRC) wird eine Linie zugefügt.



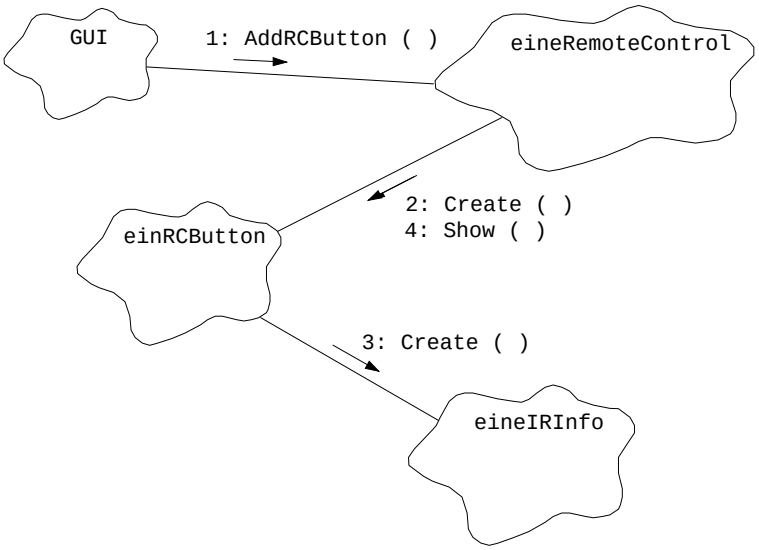
3.4.4 Objektszenario Text erstellen

In der aktuellen Fernsteuerung (SRC) wird ein neuer Text eingefügt und angezeigt.



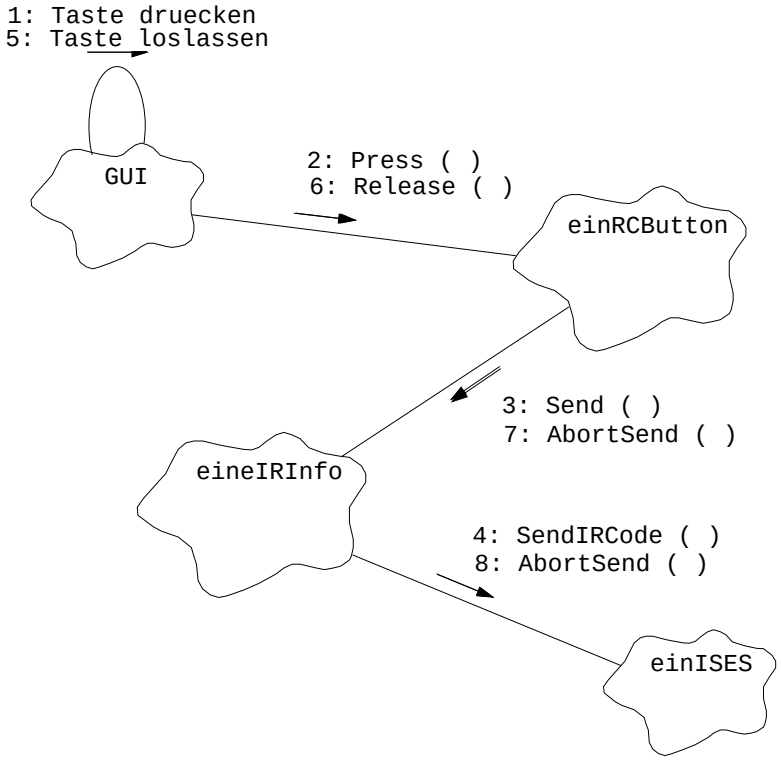
3.4.5 Objektszenario Button erstellen

In der aktuellen Fernsteuerung (SRC) wird ein neuer Button eingefügt und angezeigt.



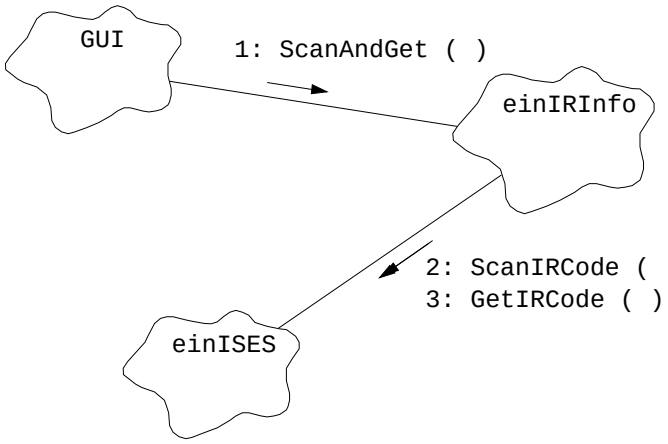
3.4.6 Objektszenario IRCode senden

Beim drücken eines Buttons wird der entsprechende IR - Code über ISES an das Zielgerät gesandt. Beim loslassen das Buttons wird die Übertragung abgebrochen.



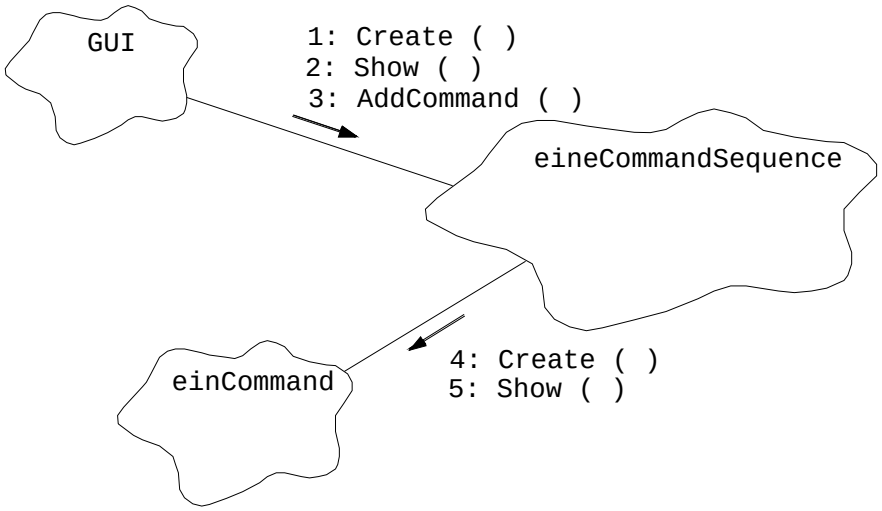
3.4.7 Objektszenario IRCode einscannen

Scannt den IR-Code von einer auf ISES gerichteten Fernsteuerung (HRC), analysiert ihn und speichert ihn im IRInfo Objekt.



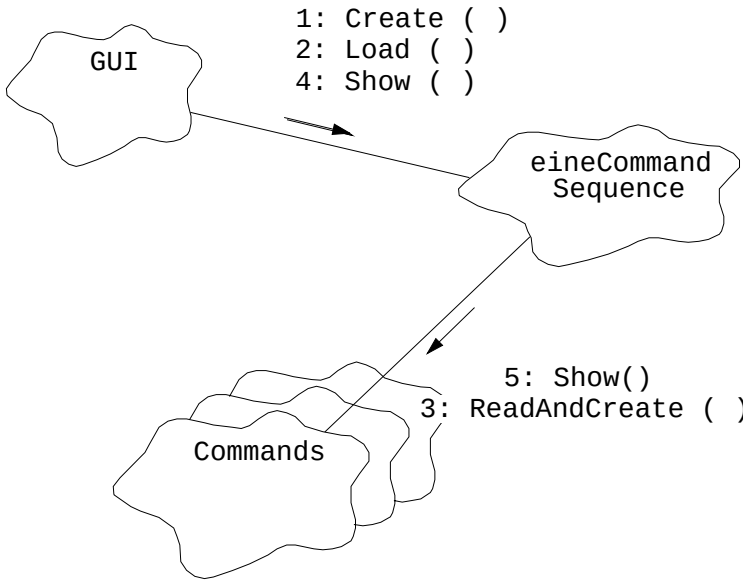
3.4.8 Objektszenario Sequenz erstellen

Der Benützer erzeugt eine neue Sequenz und fügt ihr ein Befehl zu.



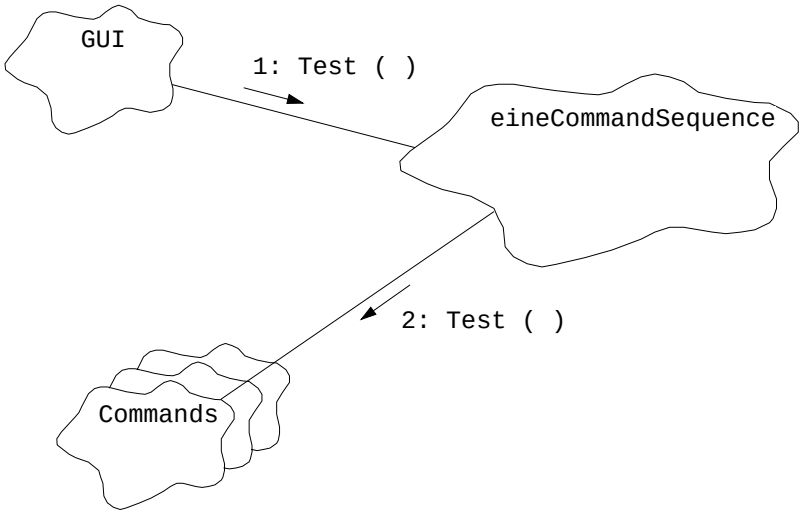
3.4.9 Objektszenario Sequenz laden

Eine in einer Datei gespeicherte Sequenz wird geladen.



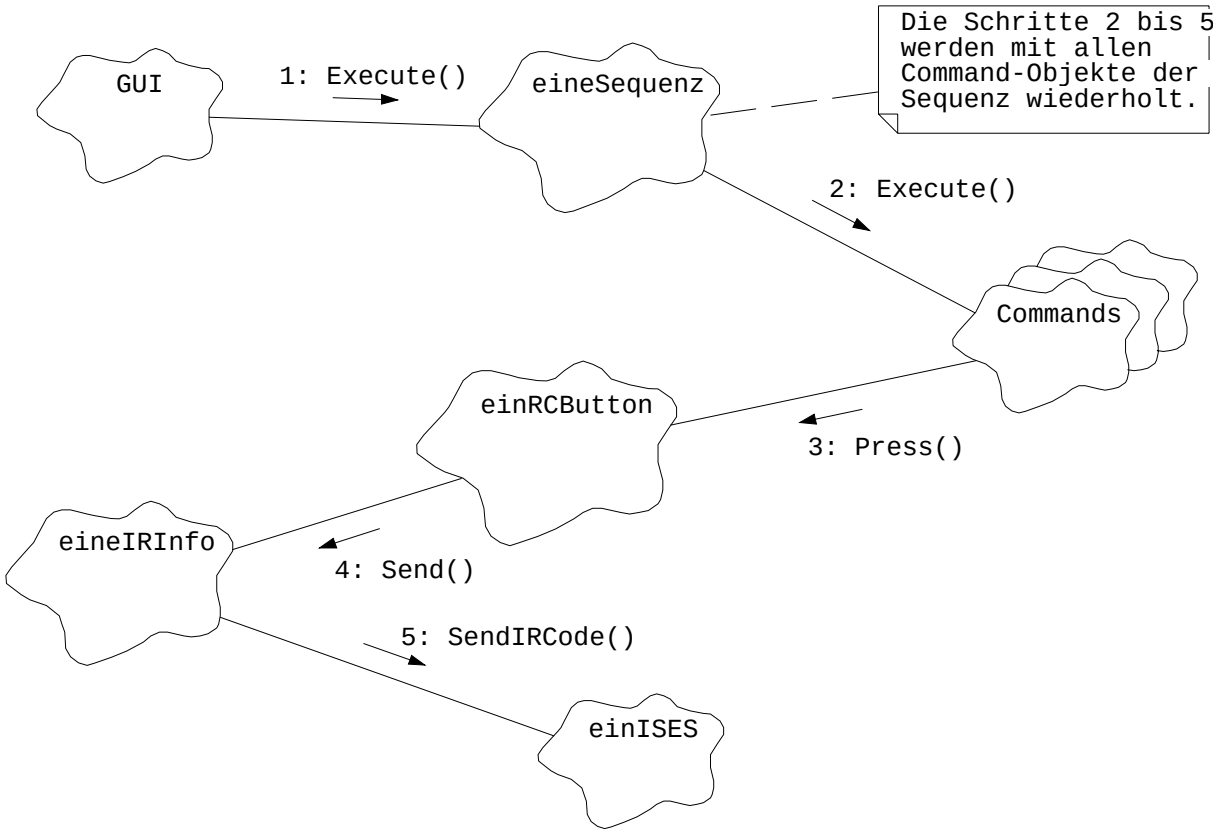
3.4.10 Objektszenario Sequenz testen

Eine Sequenz wird auf die Korrektheit seiner Befehle getestet.



3.4.11 Objektszenario Sequenz ausführen

Führt eine Sequenz mit mehreren 'PressButton ...' Befehlen aus.



3.4.12 Objektszenario Sequenz mit Wait ausführen

Eine Sequenz mit einem 'Wait' und einem 'PressButton' Befehl wird ausgeführt. Von der GUI kommt der Befehl, die Sequenz auszuführen, worauf der erste Befehl (Wait) ausgeführt wird. 'Wait' startet den Timer der Sequenz. Zu diesem Zeitpunkt wird die Abarbeitung der Sequenz abgebrochen und andere Aufgaben können erledigt werden. Wenn der Timer abgelaufen ist, wird die Sequenz angewiesen (Continue), beim nächsten Befehl weiterzufahren.