

Handbuch Konfigurationskomponente

16.04.2011

1 Einleitung

Die Konfigurationskomponente erleichtert dem Forscher die Konfiguration der Anwendung zur Extrahierung von Merkmalen. Sie bietet semantische Unterstützung und bessere Übersichtlichkeit bei der Erstellung eines Konfigurationscripts.

1.1 Überblick

Das Anwendungsfenster enthält fünf Bereiche. Abbildung 1 zeigt das Anwendungsfenster (die Darstellung des Konfigurationscripts ist aus Übersichtsgründen ausgeblendet).

Auf der linken Seite befindet sich die *Toolbox*. In diesem Fenster befinden sich die Werkzeuge zum Erstellen des Scripts. Sie sind in klappbaren Gruppen organisiert.

Auf der unteren Seite sehen Sie die *Fehlerliste*. In diesem Fenster werden Fehler und Warnungen der Anwendung ausgegeben.

Auf der rechten Seite oben befindet sich der *DSL Explorer*. In diesem Fenster wird das Script in einer Baumstruktur dargestellt. Hier kann das Script auch bearbeitet werden, indem Elemente gelöscht oder hinzugefügt werden.

Unter dem DSL Explorer ist das *Eigenschaftenfenster*. In diesem Fenster werden Eigenschaften von im Script markierten Objekten angezeigt und können bearbeitet werden.

In der Mitte des Anwendungsfensters ist der Bereich, wo das Konfigurationscript angezeigt wird und bearbeitet werden kann.

Zusätzlich befinden sich oben einige Werkzeuge zum Öffnen und Speichern von Dateien, Rückgängig machen und Wiederholen von Aktionen, sowie zum Kopieren und Einfügen.

Jeder dieser Bereiche kann beliebig positioniert und in der Größe verändert werden.

2 Arbeiten mit der Konfigurationskomponente

In den Folgenden Abschnitten wird beschrieben wie Sie die Schritte in einem typischen Arbeitsablauf mit der Konfigurationskomponente durchführen außerdem werden die verschiedenen Elemente und Werkzeuge der Anwendung erklärt.

2.1 Laden und Anlegen

Um ein neues Script anzulegen, navigieren Sie zum Menü *Datei* → *Neu* → *Datei...* und wählen Sie die Vorlage *GMAlgorithmConfigurator*. Die Anwendung legt zwei Dateien an und öffnet sie. Die Datei mit der Endung *.gmc* enthält das eigentliche Script. Die Datei mit der Endung *.gmc.diagramm* enthält Informationen über das Layout des Scripts im Editor.



Abbildung 1: Programmfenster ohne geladenes Script.

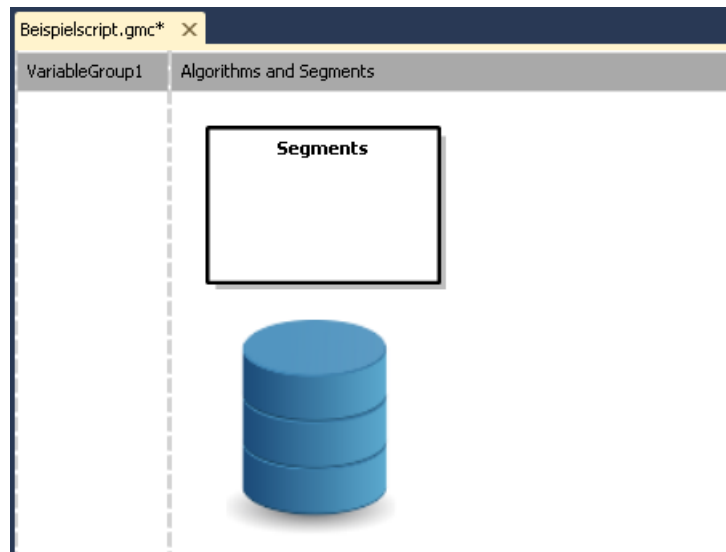


Abbildung 2: Leeres Script.

Der Verlust der .gmc.diagramm Datei führt nur zum Verlust der Anordnung der Elemente, alle Elemente und Verbindungen bleiben erhalten. In diesem Fall legt die Anwendung automatisch eine neue Layoutdatei an und ordnet die Elemente zufällig an.

Klicken Sie auf das Symbol *Speichern*, und wählen Sie einen Speicherort, um das Script zu speichern.

Um ein Script in der Anwendung zu öffnen, navigieren Sie zum Menü *Datei* → *Öffnen* → *Datei...* und öffnen Sie die Datei mit der Endung *.gmc*. Das Script wird geladen und angezeigt. Falls beim Laden ein Fehler auftritt, können Sie den Fehler in der Fehlerliste sehen.

2.2 Script Bearbeiten

Sie beginnen mit einem leeren Konfigurationsscript. Die Anwendung legt dabei automatisch einige Elemente an. Abbildung 2 zeigt das Beispielscript nach dem Laden.

Der Editorbereich ist in zwei „Schwimmbahnen“ geteilt. *VariableGroup1* ist eine Variablengruppe. *Algorithms and Segments* ist der Bereich, der die Hauptelemente des Konfigurationsscripts enthält. In der „Schwimmbahn“ *Algorithms and Segments* befindet sich ein Rechteck mit dem Namen *Segments*. Dieses Element enthält die Konfiguration der Segmente. Außerdem ist darunter ein blauer Zylinder. Dieses Element repräsentiert die Datenbank.

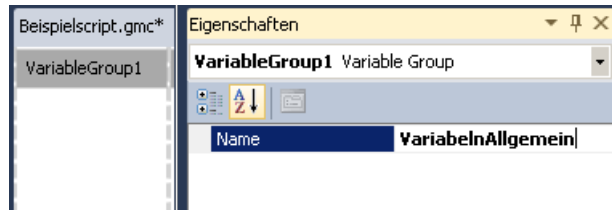


Abbildung 3: Variablengruppe umbenennen.

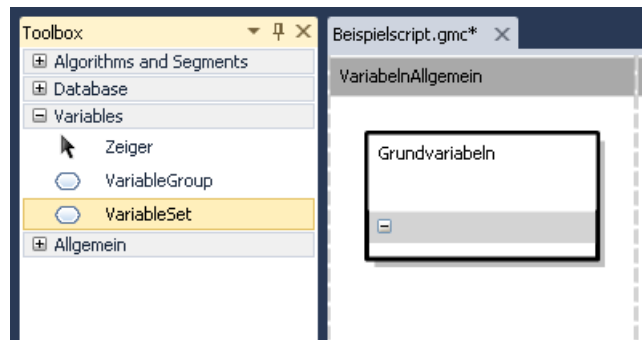


Abbildung 4: Variablenmenge hinzufügen.

2.2.1 Variablen

Wir benennen `VariableGroup1` in „`VariabelnAllgemein`“ um. Wir selektieren dazu `VariableGroup1` und ändern den Namen im Eigenschaftensfenster (Abbildung 3). Sie können weitere Variablengruppen hinzufügen, indem Sie sie aus der Toolbox auf einen leeren Bereich des Editors ziehen (außerhalb der „Schwimmbahnen“).

Wir fügen eine `VariableSet` zur Variablengruppe hinzu. Dazu ziehen wir eine `VariableSet` aus der Toolbox auf die Variablengruppe und benennen sie in „`Grundvariablen`“ um (Abbildung 4). Eine `VariableSet` ist eine Variablenmenge, sie gruppiert einzelne Variablen.

Als nächstes fügen wir der Variablenmenge eine `Variable` hinzu. Dazu klicken wir mit der rechten Maustaste auf die Variablenmenge und navigieren zu *Hinzufügen* → *Variable*. Wir benennen die `Variable` in `Anzahl` um und tragen bei *Value* den Wert 100 ein (Abbildung 5). Der Name wird automatisch mit dem Präfix `var_` ergänzt. Ein `Variable` kann benutzt werden, um Algorithmenparameter zu konfigurieren. Das ermöglicht ein Austauschen der Werte bei der Ausführung des Scripts ohne das Script ändern zu müssen.

2.2.2 Verbinden des Scripts mit Algorithmen

Um mit dem Editor arbeiten zu können, müssen sie ihm zuerst den Speicherort der Assemblies mitteilen, die Algorithmen enthalten, mit denen Sie arbeiten möchten.

Klicken Sie dazu auf eine freie Stelle im Editorfenster außerhalb der „Schwimmbahnen“ (verschieben Sie dazu die Ansicht weiter nach rechts). Suchen Sie das Eigenschaftensfenster

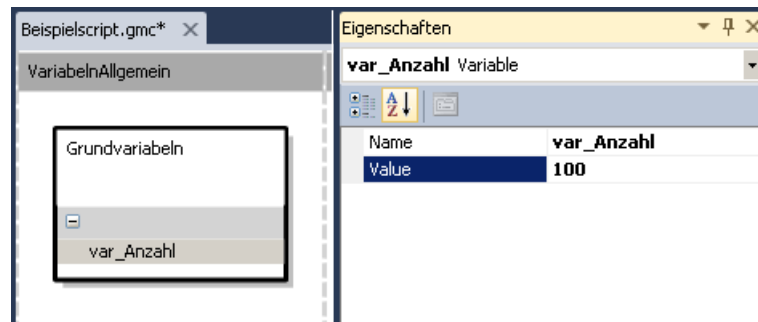


Abbildung 5: Variable hinzufügen.

(Wenn Sie es nicht sehen können, öffnen Sie es durch einen Druck auf die Taste *F4*. Unter der Überschrift *Assemblies* sind dort Eintragungen *Algorithm Assembly 1 .. 5*. Markieren Sie eine Eintragung und klicken Sie auf den Button mit den drei Punkten rechts davon. Suchen Sie nun die Assembly und öffnen Sie sie.

Der Editor analysiert die von ihnen gewählte Assembly und lädt Informationen über die Algorithmen. Wenn der Editor beim Laden einer Assembly auf fehlerhafte Informationen stößt, gibt er eine Warnung in der Fehlerliste aus.

Die Informationen über die Algorithmen werden nicht mit dem Script gespeichert, sondern bei jedem Laden des Scripts erneut eingelesen. Wenn die Anwendung eine oder mehrere Algorithmen nicht finden kann, werden entsprechende Fehlermeldungen in der Fehlerliste angezeigt und das Script kann nicht korrekt bearbeitet werden.

2.2.3 Algorithmen

Nachdem die Informationen über Algorithmen geladen wurden, können wir einen Algorithmus hinzufügen. Ziehen Sie dazu einen *Algorithm* aus der Toolbox auf die *Algorithms and Segments* „Schwimmbahn“. Daraufhin erscheint ein Fenster dass die Auswahl eines Namespace und eines Algorithmus ermöglicht (Abbildung 6).

Abbildung 7 zeigt die Darstellung des Algorithmus im Editor. Er enthält *Eingabeports* (grün), *Ausgabeports* (rot) und *Datenports* (gelb).

Eingabeports dienen dazu dem Algorithmus Daten zu übergeben. Ausgabeports sind Daten, die der Algorithmus ausgibt. Datenports repräsentieren Daten, die nach der Ausführung des Algorithmus in der Datenbank gespeichert werden sollen.

Der Algorithmus enthält außerdem *Properties*, das sind die Parameter des Algorithmus, sowie *Extremities*, hier werden Extremitäten definiert, mit denen der Algorithmus arbeiten soll.

Als Nächstes definieren wir Segmente für den Algorithmus. Zuerst selektieren wir Segments und tragen in den Eigenschaften unter Output „seg“ ein. Danach selektieren wir den Algorithmus und wählen in den Eigenschaften unter Segment „seg“ aus (Abbildung 8).

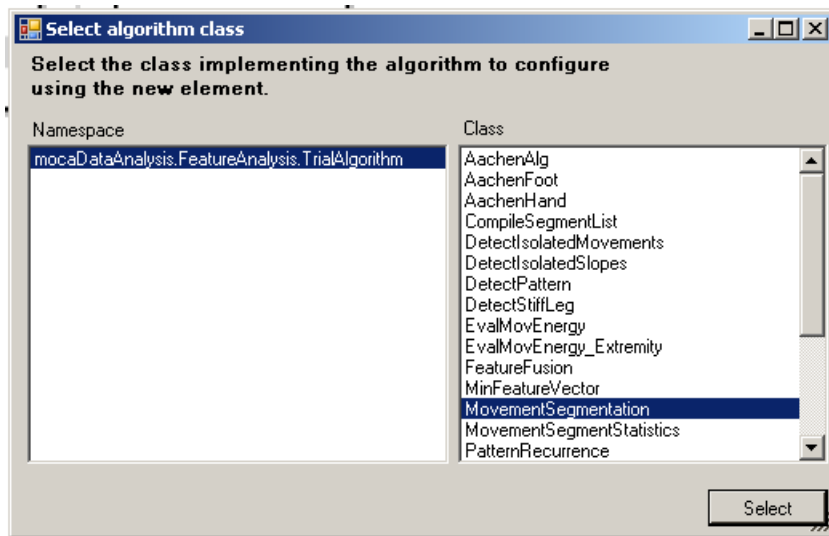


Abbildung 6: Algorithmus auswählen.

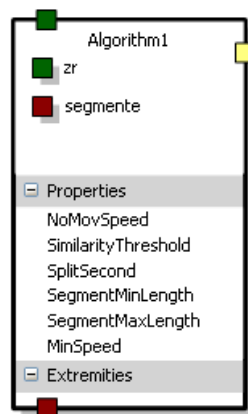
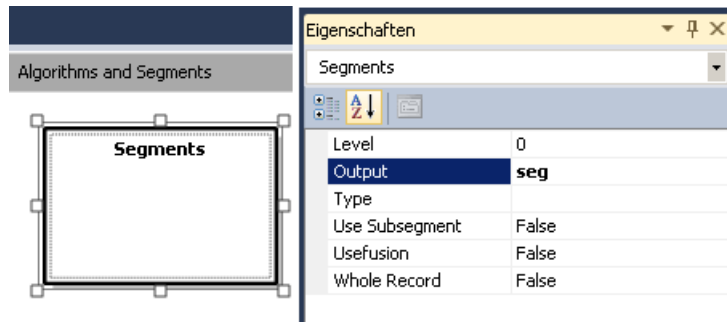
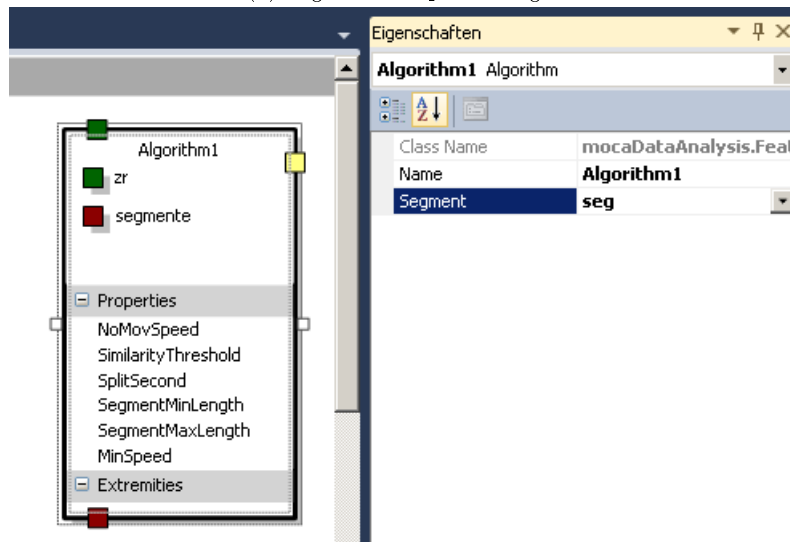


Abbildung 7: Algorithmus.



(a) Segment Output festlegen.



(b) Segmente beim Algorithmus eintragen.

Abbildung 8: Segmente definieren.



Abbildung 9: Properties hinzufügen.

Jetzt fügen wir dem Algorithmus ein *Extremity* hinzu. Dazu klicken wir mit der rechten Maustaste auf die Überschrift *Extremities* im Algorithmus und wählen „neues Extremity hinzuzufügen“. Dann nennen wir die Extremity „Arm“.

Jetzt versuchen wir eine *Property* hinzuzufügen. Dazu gehen wir analog zu Extremities vor, doch es existiert kein Menüpunkt zum Anlegen von Properties. Das liegt daran dass die Anwendung bereits alle definierten Properties automatisch angelegt hat. Sie können keine Properties für einen Algorithmus anlegen, die nicht definiert sind.

Wir müssen zuerst einige Properties löschen. Wenn wir jetzt versuchen eine Property anzulegen, erscheint ein Fenster, wo wir die Property auswählen können (Abbildung 9). Wenn das Fenster nicht erscheint, dann stand nur eine Property zur Auswahl, diese wird automatisch hinzugefügt.

Um den Wert einer Property festzulegen, selektieren wir zuerst die Property. Im Eigenschaftfenster unter *Value* können wir entweder den Wert direkt eintragen oder eine zuvor definierte Variable wählen. Abbildung 10 veranschaulicht das.

Übrigens wird im Eigenschaftfenster unter Type der Datentyp der Property angezeigt. Sie sollten prüfen, ob der eingegebene Wert diesem Typ entspricht.

2.2.4 Ports

Sie können einem Algorithmus einen Port hinzufügen, indem Sie den Port aus der Toolbox auf den Algorithmus ziehen. Die Toolbox enthält InputPort, OutputPort und DataPort, sie alle werden auf die gleiche Weise hinzugefügt. Sie können nur einen Port erstellen, wenn es noch nicht hinzugefügte Ports gibt oder wenn einer oder mehrere Ports mehrfach hinzugefügt werden dürfen. Ob ein Port mehrfach hinzugefügt werden darf erfahren Sie, wenn Sie den Port selektieren und im Eigenschaftfenster den Wert von *Allow Multiple* betrachten. Steht dort *true*, darf der Port mehrfach hinzugefügt werden.

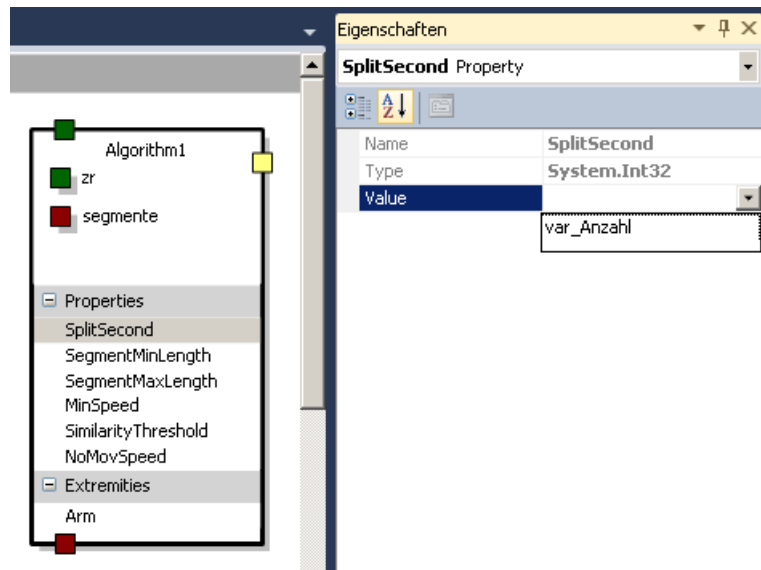


Abbildung 10: Wert der Property festlegen.

Wenn mehrere Ports zur Auswahl stehen wird ein Auswahlfenster angezeigt (Abbildung 11) ansonsten wird analog zu den Properties der einzig verfügbare Port hinzugefügt.

Sie können Ports auch löschen. Dann werden Verbindungen, die an diesen Ports enden oder von ihnen ausgehen automatisch mitgelöscht. Einige Ports können nicht gelöscht werden. Diese Ports sind an andere Ports gebunden und werden mit ihnen zusammen automatisch angelegt und gelöscht. Ein Port der andere Ports bindet hat den Wert *true* bei seiner Eigenschaft *Has Output*. Ein Port, der an einen anderen gebunden ist hat den Wert *true* bei seiner Eigenschaft *Is Attached*.

2.2.5 Verbindungen

Verbindungen können zwischen Ausgabeports und Eingabeports sowie zwischen Ports und anderen Elementen existieren

Wir wollen einen Eingabeport mit einer Zeitreihe verbinden. Dazu legen wir zuerst die Zeitreihe an, indem wir eine *Timeline* aus der Toolbox auf die *Algorithms and Segments* „Schwimmbahn“ ziehen. Dann klicken wir in der Toolbox auf das Werkzeug *TimelineConnection*, klicken auf die Timeline und dann auf den Eingabeport, den wir verbinden können. Abbildung 12 zeigt das Resultat.

Sie können die Elemente nur verbinden, wenn der Eingabeport einen Datentyp hat, der eine Zeitreihe akzeptiert. Den Datentyp des Ports sehen Sie im Eigenschaftenfenster unter *Type*.

Nun verbinden wir die Ports einiger Algorithmen miteinander. Dazu klicken wir auf das *PortConnection* Werkzeug in der Toolbox, klicken auf einen Ausgabeport und dann auf

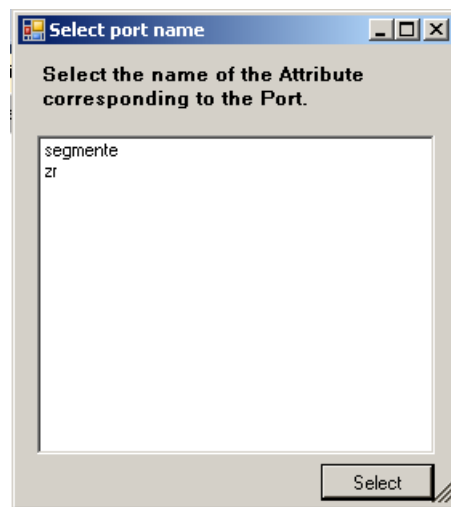


Abbildung 11: Port auswählen.

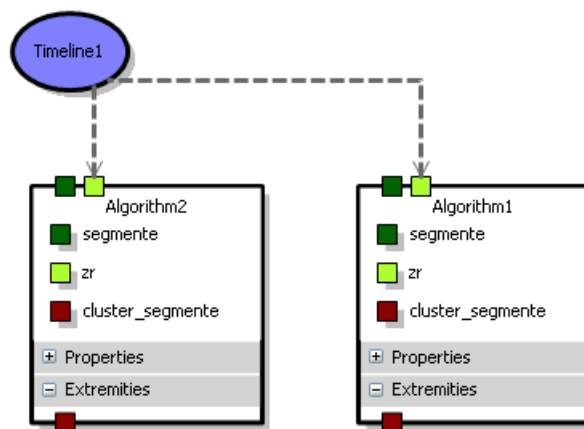


Abbildung 12: Verbindung zwischen Zeitreihe und Eingabeport.

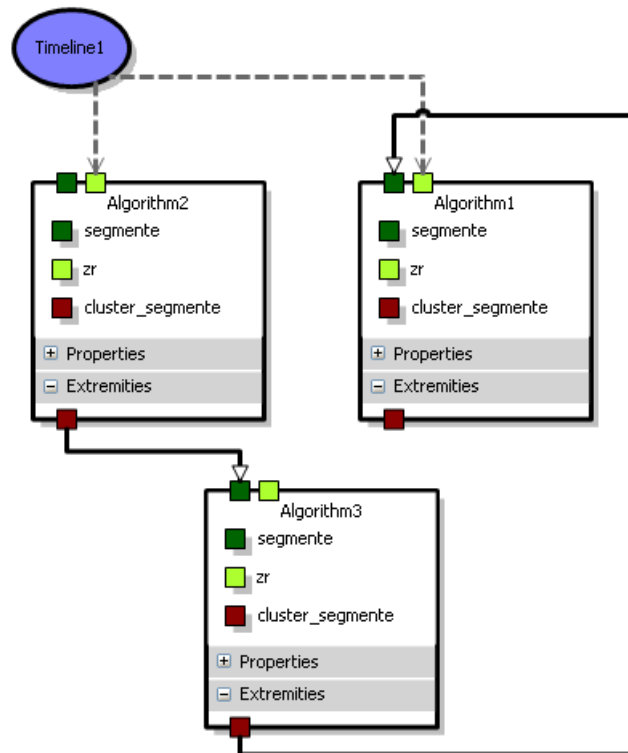


Abbildung 13: Verbindung zwischen Ports von Algorithmen.

einen Eingabeport eines anderen Algorithmus. Abbildung 13 zeigt zwei miteinander verbundene Algorithmen.

Sie können die Ports nur verbinden, wenn ihre Datentypen übereinstimmen. Ferner können Sie zwei Ports nicht verbinden, wenn die Verbindung zu einem Zyklus in den Verbindungen führen würde. Zum Beispiel können Algorithm1 und Algorithm2 in Abbildung 13 nicht verbunden werden, solange die Verbindung von Algorithm2 über Algorithm3 zu Algorithm1 besteht.

Als letztes verbinden wir noch den DataPort des Algorithmus mit der Datenbank. Dazu klicken wir auf das *DataConnection* Werkzeug in der Toolbox, klicken auf einen DataPort und dann auf den Zylinder, der die Datenbank repräsentiert. Abbildung 14 zeigt das Resultat.

3 Konfiguration

Die Konfigurationskomponente wird selbst durch eine Konfigurationsdatei konfiguriert. Diese befindet sich in einem Verzeichnis, das als allgemeines Repository für programms-



Abbildung 14: Verbindung mit der Datenbank.

spezifische Daten verwendet wird, die von allen Benutzern verwendet werden¹. Auf einer deutschen Windows XP Installation ist dieses Verzeichnis *C:\Dokumente und Einstellungen\All Users\Anwendungsdaten*. Der Name der Datei lautet *GMConfiguration.xml*.

Listing 1: Inhalt der Konfigurationsdatei.

```
<?xml version="1.0" encoding="utf-8" ?>
<editor xmlns="http://www.uni-heidelberg.de/GMConfiguration">
  <converter directory="C:\BerensteinEugen\Diplomarbeit\VS_
    Projekte\Projects\Converters"/>
</editor>
```

Listing 1 zeigt den Inhalt der Konfigurationsdatei. Bisher enthält sie nur ein Element: *converter*. Sein Attribut *directory* enthält das Verzeichnis, wo die Anwendung nach Versionskonvertieren sucht.

¹Beschreibung aus Microsoft Dokumentation zu *Environment.SpecialFolder.CommonApplicationData* im .NET Framework 4.