

Eigene Geoprocessing-Tools mit ArcObjects

Peter Schär

Bundesamt für Umwelt
Abteilung Wasser

esriuserforum.ch
Fachgruppe ArcObjects
19. September 2006



Einleitung

- ▶ Ziel = Entwicklung eigener GP-Tools mit ArcObjects
- ▶ wenig Dokumentation vorhanden (Übersicht am Schluss), aber viel Potenzial
- ▶ daher: 'Licht ins Dunkel bringen'



Ausgangslage

- ▶ Berechnungen und Analysen für das BAFU-Projekt Strategie Micropoll (Mikroverunreinigungen im Abwasser)
- ▶ Berechnung und Analyse von Abflussmengen, Abwassermengen, Stofffrachten in Zusammenhang mit Kläranlagen
- ▶ Vorwiegend Netzwerkanalyse auf geometrischem Netzwerk
- ▶ Charakter der Berechnungen ist 'Geoprocessing', d.h. wiederkehrende Verarbeitungsschritte, unterschiedliche Kombinationen von Einzelfunktionen, variierende Eingabe-Parameter
- ▶ Aber: keine Geoprocessing-Tools für geometrische Netzwerke vorhanden!
- ▶ Alternativen:
 - ▶ VBA-Scripts
 - ▶ eigene ArcObjects-Komponenten (inkl. eigene Programmierung von GUI-Elementen)
 - ▶ Eigene Geoprocessing-Tools mit ArcObjects!

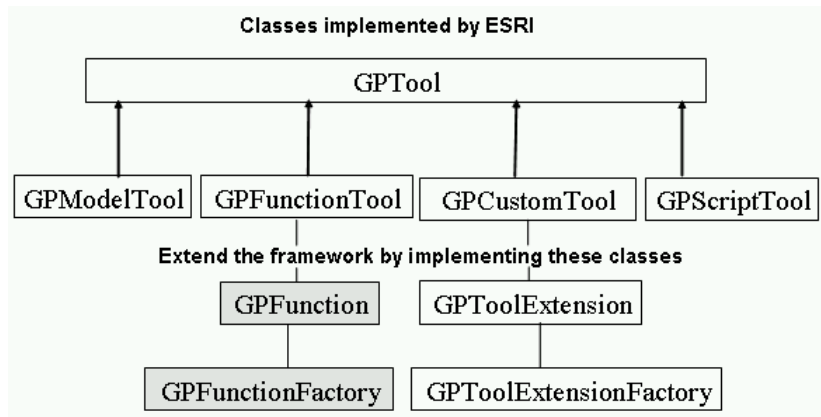


Umsetzung

- ▶ .NET-Plattform (2.0) unter Benutzung der COM-Interoperability
- ▶ C# als Programmiersprache



Geoprocessing-Tools in ArcObjects



GPFunctionFactory 1/4

- ▶ Container für mehrere Tools
- ▶ jedes Tool gehört in eine Factory
- ▶ mehrere vordefinierte Factories ('Data Management Tools', 'Analysis Tools' etc.)
- ▶ eigene Tools brauchen eigene Factory
- ▶ Interface IGPFFunctionFactory implementieren



GPFunctionFactory 2/4

IGPFunctionFactory



GPFFunctionFactory

IGPFunctionFactory : IUnknown

■ Alias: String
■ CLSID: IUID
■ Name: String

← GetFunction (in Name: String): IGPFunction
← GetFunctionEnvironments:
 IEnumGPEnvironment
← GetFunctionName (in Name: String):
 IGPName
← GetFunctionNames: IEnumGPName



GPFunctionFactory 3/4

```
1 [ComVisible(true)]
2 [Guid("83FE2101-94F2-4BFA-A9AA-29B445F5BD7F")]
3 [ClassInterface(ClassInterfaceType.None)]
4 public class MicroPoll_GPFuncFac: IGPFfunctionFactory
5 ...
6 public ESRI.ArcGIS.esriSystem.UID CLSID {
7     get {
8         UID theUID = new UIDClass();
9         theUID.Value = "MicroPoll.MicroPoll_GPFuncFac";
10
11         return theUID;
12     }
13 }
```



GPFunctionFactory 4/4

```
1 public IGPFfunction GetFunction(string Name) {
2     IGPFfunction gpFunction;
3     switch(Name)
4     {
5         case "CalculateUpstreamAccumulation":
6             gpFunction = new UpstrAcc_GPFunc();
7             break;
8         case "EstimateQ347":
9             gpFunction = new Q347_GPFunc();
10            break;
11        case "BerechneFracht":
12            gpFunction = new BerechneFracht_GPFunc();
13            break;
14        case "UebertrageFracht":
15            gpFunction = new UebertrageFracht_GPFunc();
16            break;
17    }
18    return gpFunction;
19 }
```



GPFunction 1/2

- ▶ Birgt eigentliche Funktionalität des Geoprocessing-Tools
- ▶ Zentrale Member sind:
 - ▶ ParameterInfo
 - ▶ Validate()
 - ▶ Execute()



GPFunction 2/2

IGPFunction



GPFunction

IGPFunction : IUnknown

- DialogCLSID: IUID
- DisplayName: String
- FullName: IName
- HelpContext: Long
- HelpFile: String
- MetadataFile: String
- Name: String
- ParameterInfo: IArray

- ← Execute (in paramvalues: IArray, in TrackCancel: ITrackCancel, in envMgr: IGPEnvironmentManager, in message: IGPMessages)
- ← GetRenderer (in pParam: IGPPParameter): Variant
- ← IsLicensed: Boolean
- ← Validate (in paramvalues: IArray, in updateValues: Boolean, in envMgr: IGPEnvironmentManager): IGPMessages



ParameterInfo 1/8

- ▶ Definition der Tool-Parameter (Input und Output)
- ▶ Array aus IParameterEdit-Objekten
- ▶ sehr feine/detaillierte Definition der Parameter möglich
- ▶ je präziser desto besser



ParameterInfo 2/8

IGPParameter ○
esriSystem.IClone ○
esriSystem.IPersist ○
esriSystem.IPersistStream ○
esriSystem.IXMLSerialize ○

GPPParameter

IGPParameter : IUnknown

- Altered: Boolean
- Category: String
- ChoiceList: IGPChoiceList
- ControlCLSID: IUID
- DataType: IGPDataType
- DefaultEnvironmentName: String
- Direction: esriGPPParameterDirection
- DisplayName: String
- DisplayOrder: Long
- Domain: IGPDomain
- Enabled: Boolean
- HasBeenValidated: Boolean
- Name: String
- ParameterDependencies: IEnumBSTR
- ParameterType: esriGPPParameterType
- Value: IGPValue

IGPParameterEdit ○

IGPParameterEdit : IUnknown

- Altered: Boolean
- Category: String
- ChoiceList: IGPChoiceList
- ControlCLSID: IUID
- DataType: IGPDataType
- DefaultEnvironmentName: String
- Direction: esriGPPParameterDirection
- DisplayName: String
- DisplayOrder: Long
- Domain: IGPDomain
- Enabled: Boolean
- HasBeenValidated: Boolean
- Name: String
- ParameterDependencies: IEnumBSTR
- ParameterType: esriGPPParameterType
- Value: IGPValue

← AddDependency (in Name: String)



Data Type

- ▶ lange Liste von möglichen Datentypen
- ▶ s. IGPDatatype (ArcDeveloper Help)
- ▶ DECoverageType
- ▶ DEFeatureClassType
- ▶ DEFolderType
- ▶ FieldType
- ▶ GPLongType



Domain

- ▶ Weitere Eingrenzung des Parameters innerhalb des gewählten Datentyps
- ▶ Wertebereiche
- ▶ verschiedene Arten von Geometrietypen
- ▶ Feldtypen



ParameterInfo 5/8

IGPFeatureClassDomain ○
esriGeoDatabase.IGxFilterInfo ○

GPFeatureClassDomain

IGPFeatureClassDomain : IUnknown

- Count: Long
 - FeatureType (in Index: Long): esriFeatureType
 - FeatureTypeCount: Long
 - FilterHasMs: Boolean
 - FilterHasZs: Boolean
 - FilterIndex: Boolean
 - FilterIndexType: Boolean
 - IncludeHasMs: Boolean
 - IncludeHasZs: Boolean
 - Type (in Index: Long): esriGeometryType
-
- ← AddFeatureType (in Type: esriFeatureType)
 - ← AddType (in Type: esriGeometryType)
 - ← DeleteFeatureType (in Type: esriFeatureType)
 - ← DeleteType (in Type: esriGeometryType)
 - ← MakeList (in pValue: IGPValue): IArray
 - ← SetHasMsFilter (in filter: Boolean, in IncludeHasMs: Boolean)
 - ← SetHasZsFilter (in filter: Boolean, in IncludeHasZs: Boolean)
 - ← SetIndexFilter (in filter: Boolean, in Type: Boolean)



IGPNumericDomain



GPNumericDomain

IGPNumericDomain : IUnknown

- ■ AllowEmpty: Boolean
- ■ LowLimit: Double
- ■ LowLimitInclusive: Boolean

- ← GetHighLimit (out allowHighLimit: Boolean):
Double
- ← SetHighLimit (in allowHighLimit: Boolean,
HighLimit: Double)



ParameterInfo 7/8

```
1  IGPPParameterEdit gpParam3 = new GPPParameterClass();
2  gpParam3.DataType = new GPDoubleTypeClass();
3  gpValue = new GPDoubleClass();
4  gpParam3.Value = gpValue;
5  gpParam3.Direction = _
6  esriGPPParameterDirection.esriGPPParameterDirectionInput;
7  gpParam3.DisplayName = "Inputfracht pro Einwohner und Sekunde";
8  gpParam3.Enabled = true;
9  gpParam3.Name = "input_Fracht";
10 IGPNumericDomain gpDom3 = new GPNumericDomainClass();
11 gpDom3.AllowEmpty = false;
12 gpDom3.LowLimit = 0;
13 gpDom3.LowLimitInclusive = false;
14 gpParam3.Domain = (IGPDomain) gpDom3;
15 gpParam3.ParameterType = _
16 esriGPPParameterType.esriGPPParameterTypeRequired;
17
18 parameters.Add(gpParam3);
```



ParameterInfo 8/8

```
1  IGPPParameterEdit gpParam1 = new GPPParameterClass();
2  gpParam1.DataType = new GPFeatureLayerTypeClass();
3  IGPPValue gpValue1 = new GPFeatureLayerClass();
4  gpParam1.Value = gpValue1;
5  gpParam1.Direction = _
6  esriGPPParameterDirection.esriGPPParameterDirectionInput;
7  gpParam1.DisplayName = "Ziel-Layer";
8  gpParam1.Enabled = true;
9  gpParam1.Name = "input_TargetLayer";
10 IGPPFeatureClassDomain gpDom1 = new GPFeatureClassDomainClass();
11 gpDom1.AddFeatureType(esriFeatureType.esriFTSimpleJunction);
12 gpParam1.Domain = (IGPPDomain) gpDom1;
13 gpParam1.ParameterType = _
14 esriGPPParameterType.esriGPPParameterTypeRequired;
15
16 parameters.Add(gpParam1);
```



Validate() 1/4

- ▶ Überprüfung der eingegebenen Parameter
- ▶ Test in Bezug auf die in ParameterInfo gemachten Definitionen
- ▶ Dieser Test muss nicht manuell gemacht werden, sondern mit der InternalValidate-Funktion des IGPUutilities-Interfaces.
- ▶ Fehler werden dem Benutzer automatisch angezeigt
- ▶ Validate() kann problemlos um weitere Tests erweitert werden



Validate() 2/4

```
1 public ESRI.ArcGIS.Geodatabase.IGPMessages Validate _
2 (ESRI.ArcGIS.esriSystem.IArray paramvalues, bool _
3 updateValues, IGPEnvironmentManager envMgr)
4 {
5     IGPMessages gpMsgs;
6     IGPUilities gpUtil = new GPUilitiesClass();
7
8     IArray array = this.ParameterInfo;
9     gpMsgs = gpUtil.InternalValidate(array, paramvalues, _
10 updateValues, true, envMgr);
11     return gpMsgs;
12 }
```



Validate() 3/4

```
1 private bool IsSameNetwork(IGPValue gpValue1, _
2 IGPValue gpValue2)
3 {
4     IGPUilities gpUtil = new GPUilitiesClass();
5     IFeatureClass FC1;
6     IQueryFilter QF1;
7     gpUtil.DecodeFeatureLayer(gpValue1, out FC1, out QF1);
8     INetworkClass NC1 = (INetworkClass) FC1;
9     IFeatureClass FC2;
10    IQueryFilter QF2;
11    gpUtil.DecodeFeatureLayer(gpValue2, out FC2, out QF2);
12    INetworkClass NC2 = (INetworkClass) FC2;
13
14    return NC1.GeometricNetwork.Equals(NC2.GeometricNetwork);
15 }
```



Validate() 4/4

```
1  IGPPParameter gpParam = (IGPPParameter) paramvalues.get_Element(0);
2  IGPValue gpInputValue = gpUtil.UnpackGPValue(gpParam);
3  IClone Klon = (IClone) gpInputValue;
4  IGPValue gpOutputValue = (IGPValue) Klon.Clone();
5
6  IDETable tbl = gpUtil.DecodeDETable(gpOutputValue);
7  if (tbl != null) {
8      IField fld = gpUtil.FindField(gpOutputValue, "Q347");
9      if (fld == null) {
10         IFieldsEdit fldsEdit = (IFieldsEdit) tbl.Fields;
11         IFieldEdit fldEdit = new FieldClass();
12         fldEdit.Name_2 = "Q347";
13         fldEdit.Type_2 = esriFieldType.esriFieldTypeDouble;
14         fldsEdit.AddField(fldEdit);
15         tbl.Fields = fldsEdit;
16     }
17 }
18
19 gpParam = (IGPPParameter) paramvalues.get_Element(5);
20 gpUtil.PackGPValue(gpOutputValue, gpParam);
```



Execute() 1/4

- ▶ 'Where the magic happens...'
- ▶ Funktionalität des Tools wird hier ausgeführt
- ▶ Eingegebene Parameter werden mit Utilities-Klassen (GPUutilities, DEUtilities) ausgelesen
- ▶ Parameter sind dann 'normale' Objekte (FeatureClass, NetworkClass etc.)



Execute() 2/4

```
1 private IDataset GetDataset(IGPPParameter gpParam) {
2     IDEUtilities deUtils = new DEUtilitiesClass();
3     IGPUUtilities gpUtils = new GPUUtilitiesClass();
4     IGPValue gpValue = gpParam.Value;
5     IDataset datset = deUtils.OpenDataset((IDataElement)gpValue);
6     return datset;
7 }
```



Execute() 3/4

```
1 private IField GetField(IGPPParameter gpParam) {  
2     IGPUutilities gpUtils = new GPUutilitiesClass();  
3     IGPValue gpValue = gpParam.Value;  
4     IField fld = (IField) gpValue;  
5     return fld;  
6 }
```



Execute() 4/4

```
private string GetFieldName(IGPParameter gpParam) {  
    IGPUilities gpUtils = new GPUilitiesClass();  
    IGPValue gpValue = gpParam.Value;  
    IGPString gpStr = (IGPString) gpValue;  
    string fieldName = (string) gpStr.Value;  
    return fieldName;  
}
```



Interaktion mit Benutzer

- ▶ GPMessages-Objekt steuert Interaktion mit Benutzer
- ▶ Methoden AddError, AddAbort, AddMessage (s. IGPMessages)
- ▶ einheitliche Interaktion wie alle anderen Tools (Farbgebung etc.)
- ▶ MessageBox natürlich auch möglich



Utilities-Klassen

- ▶ zur Arbeitserleichterung existieren zwei Hilfsklassen (GPUutilities, DEUtilities)
- ▶ Vielzahl an nützlichen Funktionen
- ▶ z.B. DecodeFeatureLayer o. OpenDataset (s. oben)
- ▶ viele dieser Funktionen nehmen GPValues (d.h. Parameter) entgegen und wandeln sie in 'herkömmliche' ArcObjects (FeatureClass etc.) um.
- ▶ Übersicht dazu s. IGPUutilities, IDEUtilities in der ArcDeveloper Help



Demo

Demo des Tools 'UebertrageFracht'



Fazit

Vorteile

- ▶ GP-GUI kann benutzt werden (keine eigene GUI notwendig)
- ▶ automatische Validierung des Inputs (manuell erweiterbar)

Nachteile

- ▶ schlechte Dokumentation (9.2. soll besser werden)
- ▶ wenig Infos dazu in den ESRI-Foren (geringe Verbreitung?)



Ressourcen

- ▶ EDN Documentation Library ⇒ Technical Documents ⇒ Geoprocessing ⇒ Building Geoprocessing Function Tools
- ▶ EDN Documentation Library ⇒ Library Reference ⇒ Geoprocessing ⇒ Geoprocessing Library Overview
- ▶ Thread 'Programmatically Creating Toolboxes' im Geoprocessing ArcToolbox Forum (<http://forums.esri.com/Thread.asp?c=93&f=983&t=132713>)
- ▶ Beispiele in Forum-Thread
- ▶ Technical Articles 27000 und 27680 helfen bei der Erstellung der Dokumentation des Tools



Schluss

- ▶ Herzlichen Dank für die Aufmerksamkeit
- ▶ Fragen?

