

Studiengang Systemtechnik

Vertiefungsrichtung Infotronics

Diplom 2013

Pascal Mengis

HydrOPC

Dozent Dominique Gabioud

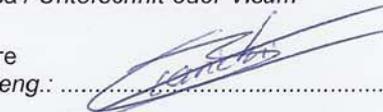
Experte Gianantonio Maniero

Sitten, 12. Juli 2013

SI	TV
X	X

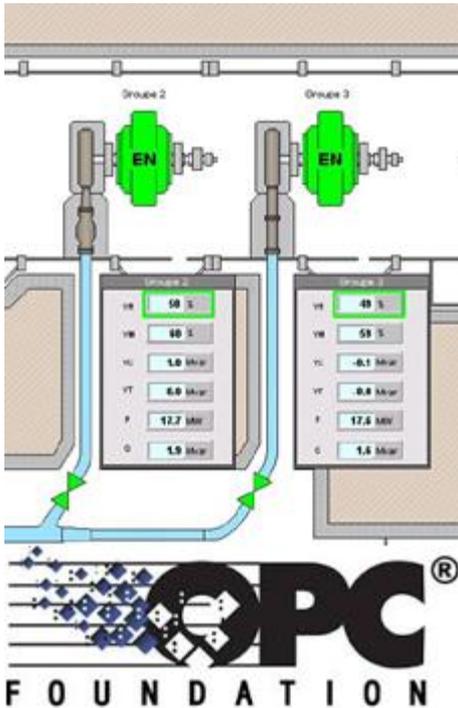
<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2012/13	No TD / Nr. DA it/2013/21
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input checked="" type="checkbox"/> Industrie HYDRO Exploitation SA <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>	Etudiant / Student Pascal Mengis	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
	Professeur / Dozent Dominique Gabioud	
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja ¹ <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) Gianantonio Maniero	

Titre / Titel <p style="text-align: center;">HydrOPC</p>
Description et Objectifs / Beschreibung und Ziele <p>Die OPC (www.opcfoundation.org) Middleware ist ein zentrales Element eines Leitsystems, da es einem Supervisor eine einheitliche Sicht eines Prozesses anbietet.</p> <p>Die Automationssysteme der Firma HYDRO Exploitation SA basieren bereits auf OPC. Die Inbetriebnahme ist jedoch nicht optimal, weil ein angepasstes Verfahren und entsprechende Werkzeuge fehlen.</p> <p>In diesem Zusammenhang sind die Ziele des Projekts HydrOPC die folgenden:</p> <ul style="list-style-type: none"> — Inbetriebnahme einer Testanlage mit SPS (Schneider, Siemens), ein IEC 61850 Schutzgerät (ABB oder Siemens), einem OPC-Server (Kepware) und einem OPC-Client (beliebiger Testclient). — Design einer Methode für die einheitliche Definition von OPC Elementen (<i>items</i>) im OPC-Server und in den Feldgeräten und Entwicklung eines entsprechenden Konfigurationsprogrammes. — Design einer Methode für die einheitliche Definition von OPC Gruppen (<i>OPCGroup</i>) im OPC-Server und im OPC-Client und, falls angemessen, Entwicklung eines entsprechenden Konfigurationsprogrammes.

Signature ou visa / Unterschrift oder Visum Resp. de la filière Leiter des Studieng.:  ¹ Etudiant/Student: 	Délais / Termine Attribution du thème / Ausgabe des Auftrags: 13.05.2013 Remise du rapport / Abgabe des Schlussberichts: 12.07.2013 Expositions / Ausstellungen Diplomarbeiten: 28 – 30.08.2013 Défense orale / Mündliche Verfechtung: Semaine Woche 36
---	---

¹ Par sa signature, l'étudiant-e s'engage à respecter strictement le caractère confidentiel du travail de diplôme qui lui est confié et des informations mises à sa disposition.

Durch seine Unterschrift verpflichtet sich der Student, die Vertraulichkeit der Diplomarbeit und der dafür zur Verfügung gestellten Informationen zu wahren.



HydrOPC

Diplomand/in Pascal Mengis

Ziel des Projekts

OPC (www.opcfoundation.org) ist ein mittlerweile weit verbreiteter Standard im Bereich der Automation und Datentransportation von Automationsnetzwerken. HydrOPC soll den Planungsprozess eines Wasserkraftwerks mit seinen speicherprogrammierbaren Steuerungen, OPC Servern und SCADA Systemen von HYDRO-Exploitation optimieren.

Methoden | Experimente | Resultate

Es wurde ein Konzept entworfen, welches unabhängig der angeschlossenen Geräte und Programmen ein Wasserkraftwerk modelliert. Die Umsetzung des Konzepts wurde mittels XML implementiert. Ein XML Dokument beschreibt die Struktur eines Wasserkraftwerks vom SCADA, bis hin zum FIELD in welchem sich die physikalischen Geräte befinden. Die entwickelte Richtlinie gewährleistet eine kohärente und fast redundanzfreie Arbeitsweise um die Prozessvariablen eines Automationsprozesses eines Wasserkraftwerks zu entwerfen oder modifizieren. Um eine vereinfachte Handhabung des SCADA Systems zu gewährleisten wurde eine OPC Client/Server Architektur verwendet. Durch die Standardisierte OPC Schnittstelle werden die Prozessdaten Protokoll unabhängig von den sich dahinter befindenden Feldprotokollen transferiert.

Mittels XSL Transformationen wurden aus dem XML Dokument basierend auf dem XSD Dokument alle Konfigurationen der gesamten Automationskette generiert. Diese Programme dienen als Referenzumgebung.

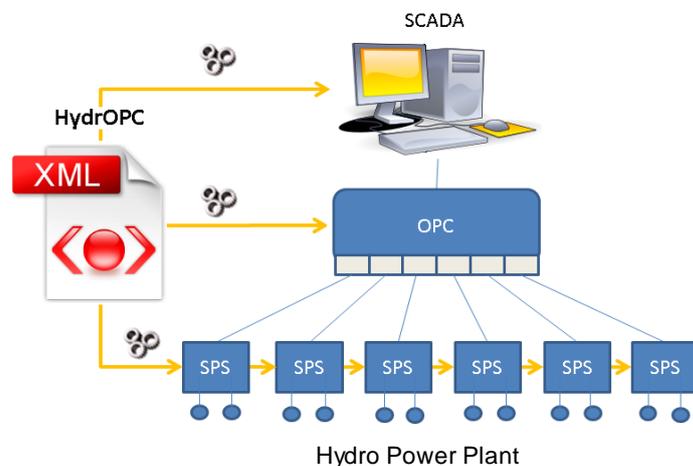
Diplomarbeit
 | 2013 |

Studiengang
 Systemtechnik

Anwendungsbereich
 Infotronik

Verantwortliche/r Dozent/in
 M. Dominique Gabioud
 dominique.gabioud@hevs.ch

Partner
 HYDRO-Exploitation



Inhaltsverzeichnis

Einleitung.....	5
SCADA Entwicklungsprozess.....	5
Architektur einer Wasserkraftwerkanlage	5
Heutiges Vorgehen	6
Ziel	7
Technologie XML, XSD, XSLT & OPC	8
XML.....	8
XSD.....	8
XSLT	9
OPC.....	9
Einleitung SCADA & FIELD	10
SCADA.....	11
UML Design	12
XSD Design.....	13
Aufbau	14
IEC61850.....	18
FIELD	21
UML Design	22
XSD Design.....	22
Aufbau	23
Beispiel einer erstellten Infrastruktur	28
Bemerkung:	29
Bemerkung:	29
Referenzumgebung & dazugehörige XSLT	30
oXygene V12.....	30
Cimplicity	31
XSLT Cimplicity.....	32
KEPServerEXV5	34
XSLT Kewpare	35
Beispiel:	38
Bemerkung:	40

Unity V7	40
XSLT Unity.....	41
TIA Portal V11.....	43
XSLT TIA Portal V11	43
Ergänzende XSLT.....	45
XSLT OPC Groups	45
XSLT Alarmclasses.....	46
XSLT Resources	47
Test & Validation	48
TIA Portal V11.....	49
Unity V7	50
Kepware.....	52
Cimplicity	53
Kepware mit den angeschlossenen Feldgeräten.....	55
Kepware mit dem Cimplicity OPC Client	56
Benutzeroberfläche	57
Schlussfolgerung.....	59
Literaturverzeichnis.....	60
Datum	60
Unterschrift	60
Anhang	61

Abbildungsverzeichnis

Abbildung 1: Architektur ohne OPC	5
Abbildung 2: Architektur mit OPC	6
Abbildung 3: Ziel von HydrOPC	7
Abbildung 4: Typische OPC Implementation [5]	10
Abbildung 5: Konzept SCADA & FIELD	10
Abbildung 6: SCADA von HYDRO-Exploitation	11
Abbildung 7: UML eines SCADA Systems	12
Abbildung 8: Beispiel eines einfachen SCADA System welches sich auf dem UML stütz	13
Abbildung 9: SCADA Abschnitt des framework.xml Dokuments.....	13

Abbildung 10: <scada> Element 14

Abbildung 11: <group> Element..... 14

Abbildung 12: <node> Element..... 15

Abbildung 13: <a> <s> <p> <m> Elemente 16

Abbildung 14: <point> Element..... 16

Abbildung 15: semantic Attribut 16

Abbildung 16: Prinzip der IEC61850 Norm..... 18

Abbildung 17: Deflektor laut IEC61850-7-410 Norm 19

Abbildung 18: Eintrag von HYDRO-Exploitation (Anhang 4: Gr3 Liste superviseur.xls) 19

Abbildung 19: Einbindung von IEC61850 Geräten 20

Abbildung 20: Architektur für FIELD Abschnitt 21

Abbildung 21: UML FIELD Architektur 22

Abbildung 22: FIELD Abschnitt des framework.xml Dokuments..... 22

Abbildung 23: <field> Element 23

Abbildung 24: <agent> Element 24

Abbildung 25: <device> Element..... 25

Abbildung 26: <address> Element..... 26

Abbildung 27: <items> Element 26

Abbildung 28: <collection> Element 26

Abbildung 29: <item> Element..... 27

Abbildung 30: framework.xml basierend auf framework.xsd..... 28

Abbildung 31: Digitale Ausgangskarte in Unity V7 30

Abbildung 32: Format eines zu Importierenden .csv Dokuments für Cimplicity 31

Abbildung 33: XSLT zur Generierung des import_points.csv Dokuments..... 32

Abbildung 34: KEPServerEX V5 Design 34

Abbildung 35: XSLT Kepware Prinzip 35

Abbildung 36: Flussdiagramm XSLT Kepware 36

Abbildung 37: XSLT Code für die Adresskonvertierung..... 37

Abbildung 38: Konversion des <address> Elements mit Trennzeichen 38

Abbildung 39: Interner Speicher referenzen Tabelle auf Modbusadresse 39

Abbildung 40: Konversion des <address> Elements ohne Trennzeichen..... 39

Abbildung 41: XSLTUnity Prinzip 41

Abbildung 42: <address> Element Trennzeichen Behandlung in Unity V7 42

Abbildung 43: TIA Portal Zellenformat der Importation 43

Abbildung 44: XSLT Sonderzeichen 44

Abbildung 45: Resultat der XSLTOPCGroup.xsl 46

Abbildung 46: Resultat der XSLTAlarmclasses..... 46

Abbildung 47: Resultat der XSLTResources..... 47

Abbildung 48: XSLT Transformationen für die entsprechenden Schichten 48

Abbildung 49: Resultat der Importation unter TIA Portal V11..... 49

Abbildung 50: Resultat der Importation unter Uinity V7 50

Abbildung 51: Interne Speicheradressierung der digitalen Eingangskarte 51

Abbildung 52: Ausführung des kepwareconfig.bat Dokuments 52

Abbildung 53: Kepware Modbusgerät DEVQuantum1 mit seinen Prozessvariablen 52

Abbildung 54: Kepware Siemensgerät DEVSiemens1 mit seinen Prozessvariablen 53

Abbildung 55: Name des Device unter Cimplicity 53

Abbildung 56: Cimplicity Importation der Points 54

Abbildung 57: Importierte Points in Cimplicity 54

Abbildung 58: TIAPortalV11 Wert der Prozessvariable TCSP_SCD_POSMAX 55

Abbildung 59: KEPServerEXV5 Wert der Prozessvariable TCSP_SCD_POSMAX 55

Abbildung 60: Unity V7 Wert der Prozessvariablen PRSP_CONT_EDN_L_M_E und PRSP_CONT_EDN_L_M_F 55

Abbildung 61: KEPServerEXV5 Werte der Prozessvariablen PRSP_CONT_EDN_L_M_E und PRSP_CONT_EDN_L_M_F 56

Abbildung 62: OPC Client in Cimplicity 56

Abbildung 63: Verbundene OPC Clients auf dem Kepware OPC Server 57

Abbildung 64: Mögliches Design für den SCADA Abschnitt 58

Abbildung 65: Mögliches Design für das Mapping der points mit den items 58

Einleitung

HYDRO-Exploitation bietet Dienstleistungen auf dem Gebiet Betrieb und Instandhaltung von Wasserkraftanlagen. Ihr Kerngeschäft liegt im Planen und Automatisieren von neuen und bestehenden Wasserkraftwerken. Dabei entwickeln und implementieren sie eine Infrastruktur mit speicherprogrammierbaren Steuerungen kurz SPS und einer Supervisionssoftware. HydrOPC soll den bestehenden Prozess der Planung, Erneuerung und Implementierung optimieren.

SCADA Entwicklungsprozess

Unter Supervisory Control and Data Acquisition (SCADA) versteht man das Überwachen und Steuern technischer Prozesse mittels eines Computer-Systems. [1] Dieser Prozess besitzt eine Datenbank in welcher: Zeit, Art, alte Werte und weitere Informationen über den zu regelnden Prozess gespeichert werden.

Architektur einer Wasserkraftwerkanlage

Die Architektur einer Wasserkraftwerkanlage kann anhand zweier Architekturen erklärt werden:

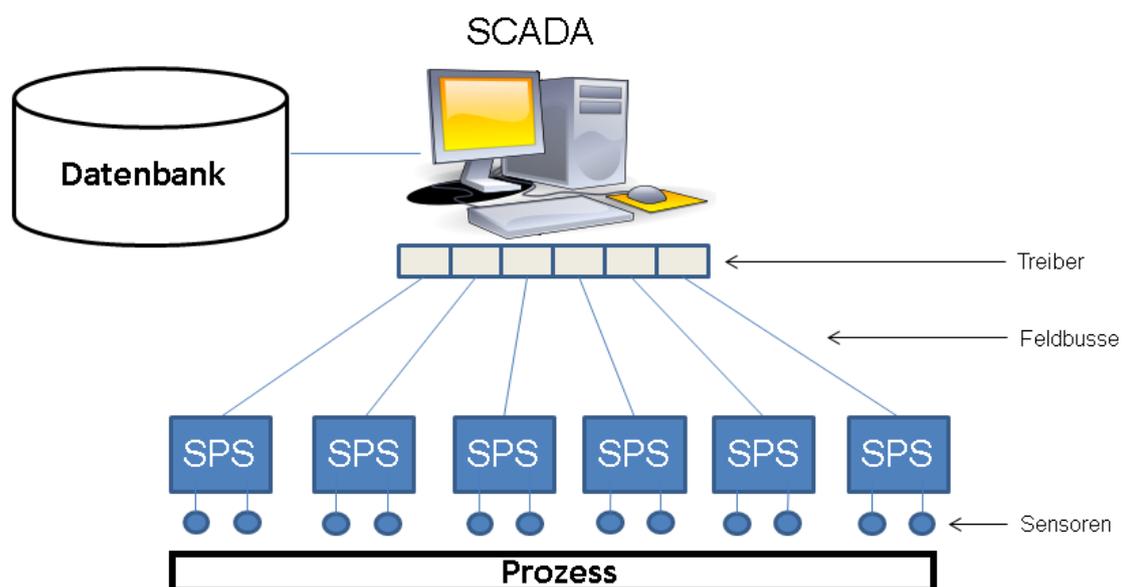


Abbildung 1: Architektur ohne OPC

Bei der Architektur ohne OPC (OLE Processing Control, Erläuterung siehe Kapitel: Technologie XML, XSD, XSLT & OPC unter OPC) muss das SCADA Programm die Kommunikation mit den sich darunter befindenden Geräten und deren Protokolle koordinieren.

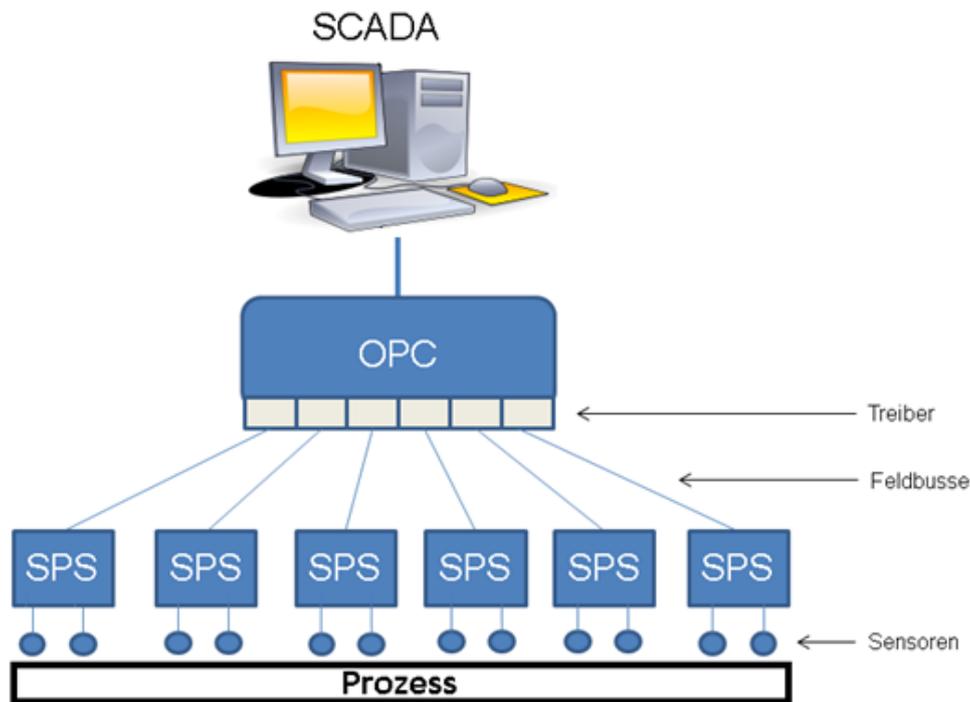


Abbildung 2: Architektur mit OPC

Der Vorteil der Architektur mit OPC ist, dass das SCADA Programm sich nicht um die spezifisch angeschlossenen Protokolle kümmern muss. Dies wird durch die OPC Schnittstelle geregelt.

HYDRO-Exploitation verwendet zurzeit mehrheitlich die Architektur ohne OPC.

Heutiges Vorgehen

Zurzeit besteht das Planen einer neuen Infrastruktur darin, Excel Dateien mit den Ein- und Ausgängen der verschiedenen SPS deren Beschreibung, Nomenklatur, Adresse usw. zu erstellen. Weitere Excel Dokumente werden benötigt um die Konfiguration der SPS zu generieren und nochmals weitere für die Konfiguration der Supervisionssoftware. Dabei ist es sehr umständlich und mühsam sich eine kohärente und einheitliche Arbeitsweise anzueignen, in welcher überall dieselben Namen der Prozessvariablen und deren Beschreibungen vorherrschen. Wird zum Beispiel ein neuer Sensor in den zu regelnden Prozess integriert, muss dieser in jedem Excel Dokument manuell hinzugefügt und anschliessend in den einzelnen Programmen implementiert werden. Diese Arbeitsweise stützt sich auf eine mittlerweile veraltete Technologie da aus den Excel Dokumenten heraus CSV (Comma-Separated Values) Dokumenten erstellt werden. Diese werden heute länger je mehr durch XML Dokumente (siehe Kapitel: XML) ersetzt, welche eine viel höhere Flexibilität besitzen.

Ziel

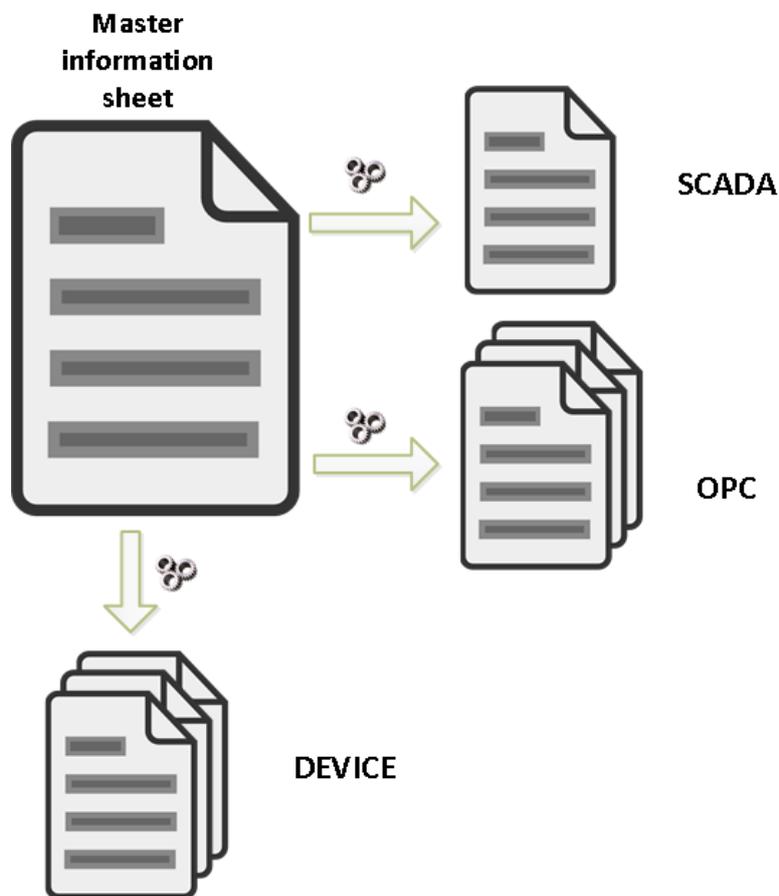


Abbildung 3: Ziel von HydrOPC

Ziel der Diplomarbeit HydrOPC ist es deshalb, ein einheitliches Verfahren zu entwickeln in welchem eine Kohärenz der Prozessvariablen durch alle Schichten (SPS, OPC und SCADA) herrscht und in nur einem Dokument vereinheitlicht wird. Es soll eine Richtlinie entwickelt werden, welche beschreibt wie das Dokument alle Informationen einer zu planenden oder bestehenden Infrastruktur besitzt, auszusehen hat. Aus diesem Dokument sollen alle gewünschten Konfigurationen der verschiedenen Schichten extrahiert werden können.

Die Richtlinie soll:

- eine einheitliche Definition von OPC Elementen (items) im OPC Server und in den Feldgeräten beinhalten.
- eine einheitliche Definition von OPC Gruppen (OPCGroup) im OPC Server und im OPC-Client beinhalten.

Um die Entwickelte Richtlinie zu testen soll eine Testanlage in Betrieb genommen werden. Die Testanlage besteht aus:

- eine SPS von Siemens (S7-300)

- eine SPS von Schneider (Quantum)
- ein IEC61850 Schutzgerät
- einem OPC Server von Kepware (KEPServerEXV5)
- einem OPC Client von einem beliebigen Hersteller

Das IEC61850 Schutzgerät wurde im Verlauf der Diplomarbeit aus Zeitmangel aus der Testanlage und der Richtlinie entfernt.

Technologie XML, XSD, XSLT & OPC

XML

Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. XML wird u. a. für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt, insbesondere über das Internet. [2]

Die Vorteile dieser Sprache liegen darin, dass sie leicht verständlich und trotz spezieller Syntax leserlich bleibt. Sie bietet sehr viele Möglichkeiten im Bereich der Strukturierung von Daten innerhalb einer Textdatei. XML ist leicht zu erlernen, weil sie einen ähnlichen und verständlichen Aufbau wie HTML besitzt.

XSD

XML Schema, abgekürzt XSD (XML Schema Definition), ist eine Empfehlung des W3C zum Definieren von Strukturen für XML-Dokumente. Anders als bei den klassischen XML-DTDs wird die Struktur in Form eines XML-Dokuments beschrieben. Darüber hinaus wird eine große Anzahl von Datentypen unterstützt.

Das XML Schema beschreibt in einer komplexen Schemasprache Datentypen, einzelne XML-Schema-Instanzen (Dokumente) und Gruppen solcher Instanzen. Ein konkretes XML Schema wird auch als eine XSD (XML Schema Definition) bezeichnet und wird als Datei üblicherweise durch die Endung „.xsd“ ersichtlich. Im Gegensatz zu DTDs kann bei Verwendung von XML Schemata zwischen dem Namen des XML-Typs und dem in der Instanz verwendeten Namen des XML-Tags unterschieden werden. [3]

Ein Vorteil des XML Schemas XSD liegt darin, dass eine Richtlinie geschrieben werden kann, welche die Struktur der darauf aufbauenden XML Dokuments vorgibt. Eine XSD ist zu vergleichen mit der Grammatik einer Sprache. Falsch geschriebene oder fehlerhafte Einträge in einem XML Dokument, welches auf einer XSD basiert, werden gekennzeichnet.

XSLT

XSL Transformation, kurz XSLT, ist eine Programmiersprache zur Transformation von XML-Dokumenten. Sie ist Teil der Extensible Stylesheet Language (XSL) und stellt eine turing-vollständige[2] Sprache dar.

XSLT baut auf der logischen Baumstruktur eines XML-Dokumentes auf und dient zur Definition von Umwandlungsregeln. XSLT-Programme, sogenannte XSLT-Stylesheets, sind dabei selbst nach den Regeln des XML-Standards aufgebaut.

Die Stylesheets werden von spezieller Software, den XSLT-Prozessoren, eingelesen, die mit diesen Anweisungen ein oder mehrere XML-Dokumente in das gewünschte Ausgabeformat umwandeln. XSLT-Prozessoren sind auch in vielen modernen Webbrowsern integriert, wie zum Beispiel Opera (ab Version 9), Firefox und Internet Explorer Version 5 (erst seit Version 6 mit vollständiger XSLT-1.0-Unterstützung). [4]

Der Vorteil einer XSLT liegen darin, dass aus jedem beliebigen XML Dokument heraus ein anderes Textbasierendes Dokument erstellt werden kann. Dieses kann dabei Informationen aus dem XSL Dokument enthalten wie auch aus dem zu transformierenden XML Dokument.

OPC

OLE for Process Control kurz OPC (www.opcfoundation.org) ist eine Software-Schnittstelle, die den Datenaustausch zwischen Anwendungen und Geräten verschiedener Hersteller im Bereich der Automation ermöglichen soll. OPC wird häufig in Automationsnetzwerken verwendet in welchen Gerätschaften verschiedener Hersteller verwendet werden. Der Datenaustausch wird somit flexibel und Geräte unabhängig. Die Daten können ohne weiteres in höheren Schichten wie SCADA oder HMI (Human Machine Interface) verwendet werden.

Typical Plant Network Implementation

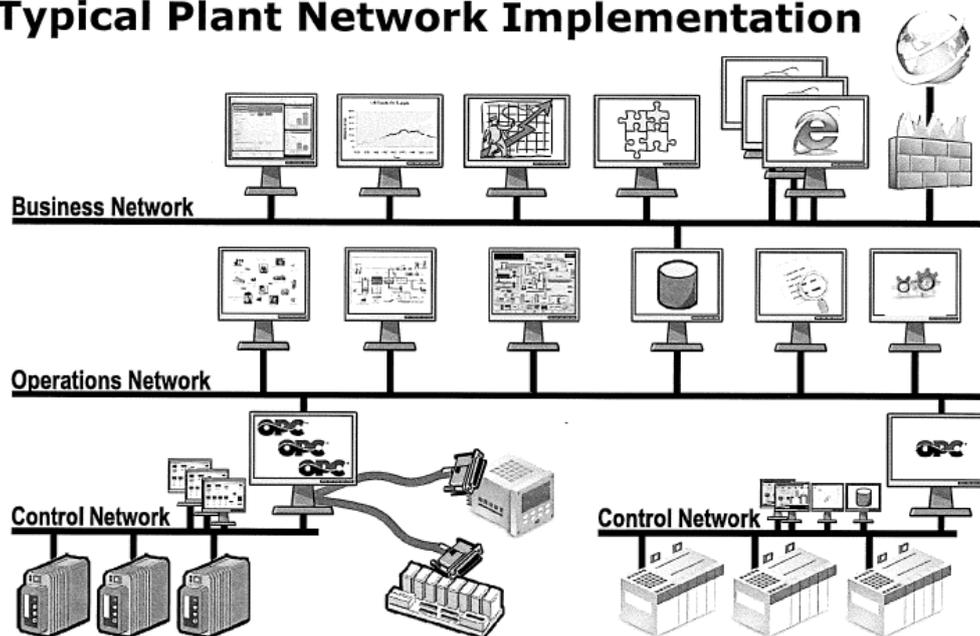


Abbildung 4: Typische OPC Implementation [5]

OPC stützt sich auf eine typische Server Client Architektur. Ein OPC Server besitzt dabei alle Prozessvariablen der sich darunter befindenden Geräte. Diese können von einem OPC Client gruppiert und gelesen werden. Die dazu verwendeten Prinzipien und Methoden können im Anhang 1: OPCBericht.pdf gefunden werden.

Einleitung SCADA & FIELD

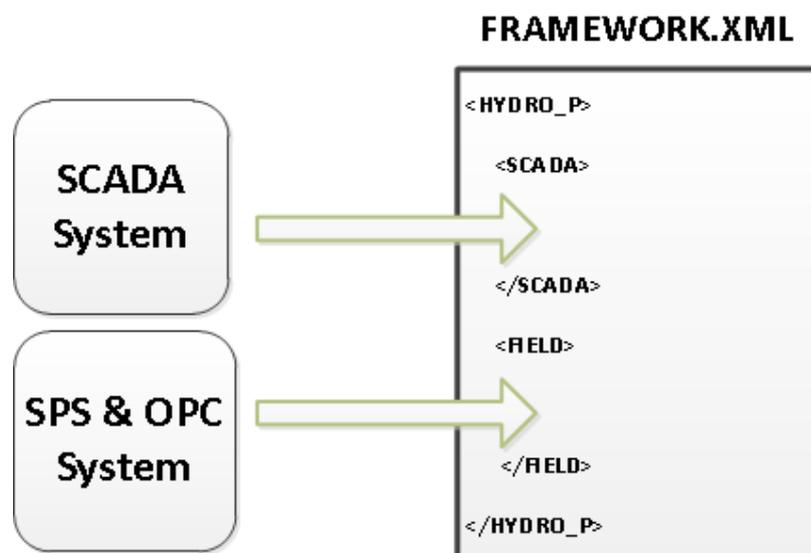


Abbildung 5: Konzept SCADA & FIELD

Die Entwicklung der einheitlichen Richtlinie kann in zwei Abschnitte unterteilt werden der SCADA und der FIELD Abschnitt. Beide befinden sich Innerhalb eines Dokuments und werden in den folgenden Kapiteln beschrieben.

SCADA

Ein Typisches SCADA System von HYDRO-Exploitation sieht folgendermassen aus:

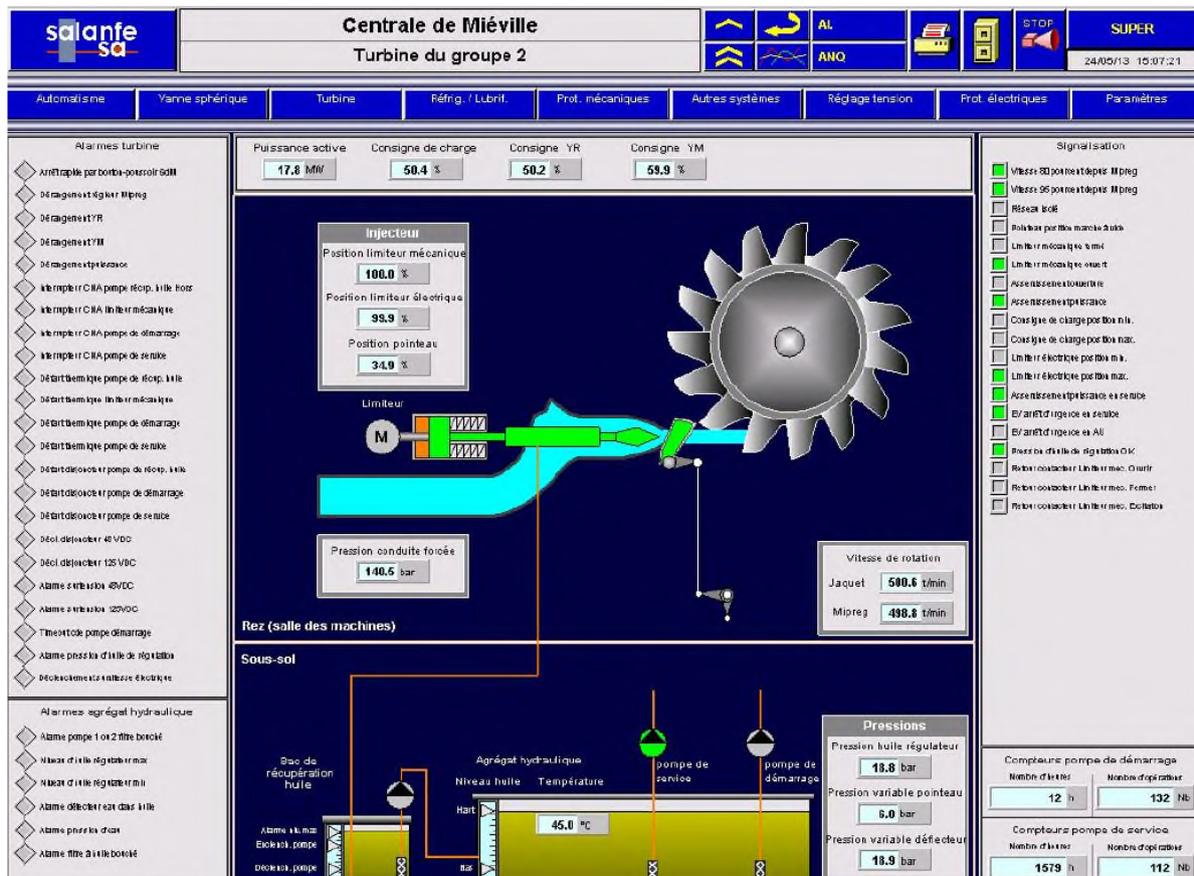


Abbildung 6: SCADA von HYDRO-Exploitation

Es handelt sich hier nur um einen Ausschnitt eines Wasserkraftwerks. Ersichtlich ist, dass alle Informationen Prozessvariablen innerhalb des SCADA Systems in verschiedene Kategorien unterteilt werden können. Die Kategorien sind folgende:

- Parameter
- Messungen
- Signalisationen
- Alarme

Die Kategorie Parameter entspricht einer Prozessvariable welche an eine bestimmte Eigenschaft eines Systems gebunden ist. Zum Beispiel gehört die maximale oder minimale Position welche eine Einspritzdüse haben kann zu dieser Kategorie.

Zur Kategorie Messungen gehören physikalische Werte wie zum Beispiel: Temperaturen, Drücke, Geschwindigkeiten usw.

Die Kategorie Signalisationen besitzt Indikationen welche die Zustände gewisser Komponenten anzeigen. Typischerweise gehören Zustände wie: Pumpe Ein/Aus , maximale oder minimale Position erreicht usw. dazu.

Zur Kategorie der Alarme gehören Indikationen welche physikalischen Schäden oder Systemausfälle des Systems anzeigen. Zum Beispiel gehören verstopfte Filter oder zu hohe Rotationsgeschwindigkeiten der Turbine zu der Alarm Kategorie.

Durch die Analyse der vier Kategorien und den sich dahinter befindenden Einträge, ersichtlich im Anhang 2: LISTimport.xlsx Dokument, wurde ein Konzept entworfen, dass unabhängig der sich dahinter befindenden Software ein SCADA Prinzip modelliert.

UML Design

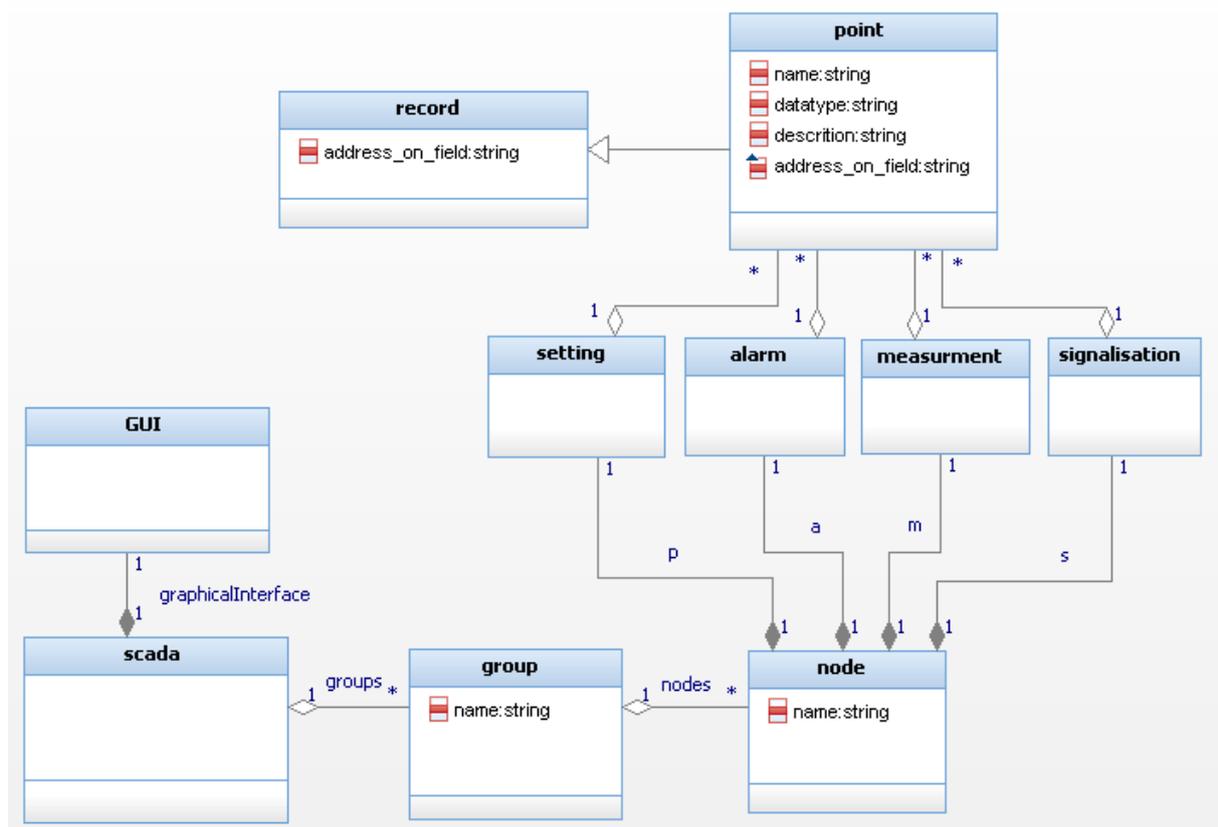


Abbildung 7: UML eines SCADA Systems

Das UML Diagramm entspricht einem sehr generellen Beispiel und dient zum globalen Verständnis wie ein SCADA System aufgebaut sein kann. Das ganze kann für eine Turbine und einen Generator folgendermassen betrachtet werden:

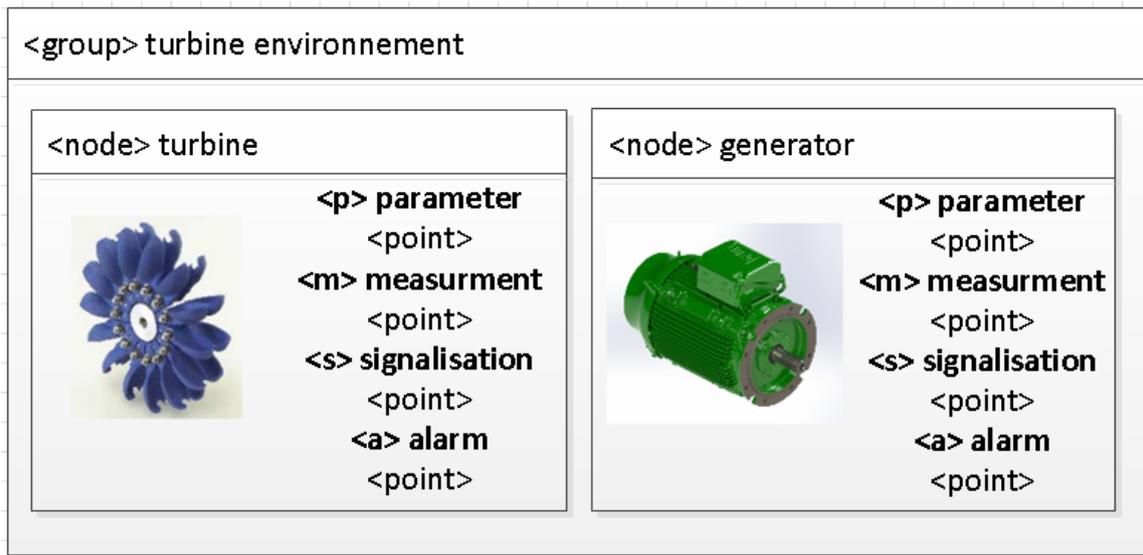


Abbildung 8: Beispiel eines einfachen SCADA System welches sich auf dem UML stütz

Wobei ein Point mit einem der sich im OPC Server oder direkt auf dem Gerät befindender Prozessvariable verbunden ist. Dabei kann die OPC Schnittstelle oder das entsprechenden Geräteprotokoll verwendet werden um die Verbindung zu gewährleisten.

XSD Design

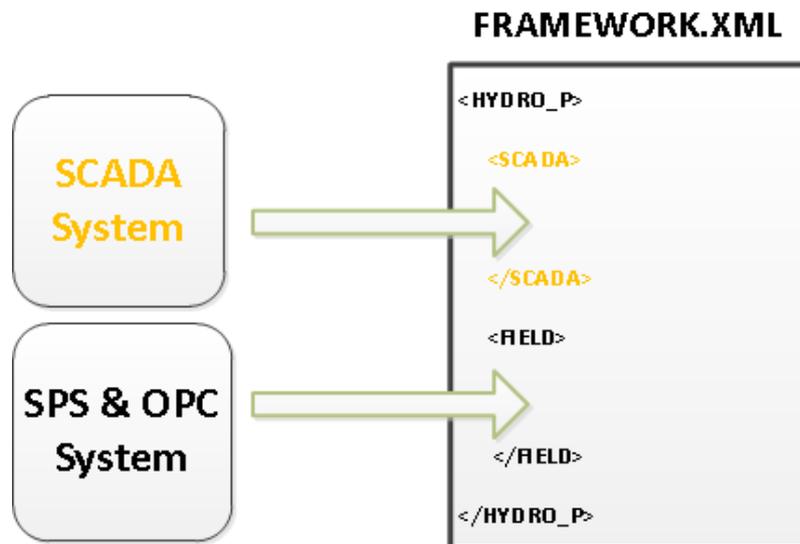


Abbildung 9: SCADA Abschnitt des framework.xml Dokuments

Das in der Abbildung 7: UML eines SCADA Systems UML kann in ein XML Schema (XSD) übersetzt werden. Das daraus resultierende Schema dient als Grundgerüst für jedes neue oder bestehende Projekt welches das SCADA System einer Wasserkraftwerk-Infrastruktur darstellen soll. Ein vereinfachtes XML Beispiel erleichtert das Verständnis des folgenden Kapitels:

```

<scada>
  <group name="GROUP1">
    <node name="NODE1">
      <p>
        <point description="A" datatype="INT" ref="ID-000001"/>
      </p>
      <m>
        <point description="B" datatype="INT" ref="ID-000002"/>
      </m>
      <s>
        <point description="C" alarm_class="CLASS2" datatype="BOOL" ref="ID-000003"/>
      </s>
      <a>
        <point description="D" alarm_class="CLASS1" datatype="BOOL" ref="ID-000004"/>
      </a>
    </node>
  </group>
</scada>

```

Aufbau

Das XML Schema ist wie folgt aufgebaut. Der Code kann im Anhang 3: framework.xsd betrachtet werden:

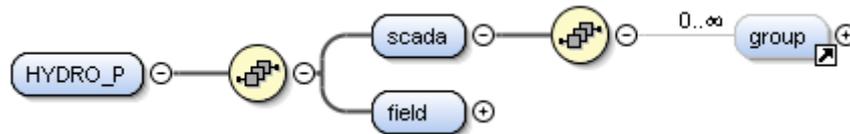


Abbildung 10: <scada> Element

Ein Projekt besitzt ein <scada> Element. Ein <scada> Element kann 0 bis unendlich viele <group> Elemente besitzen.

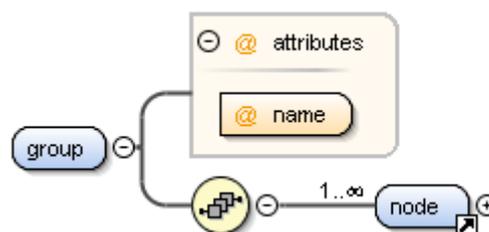


Abbildung 11: <group> Element

Ein <group> Element besitzt ein Attribut namens name, welches den Namen der gewünschten OPCGroup beinhaltet. Ein <group> Element kann 1 bis unendlich viele <node> Elemente besitzen.

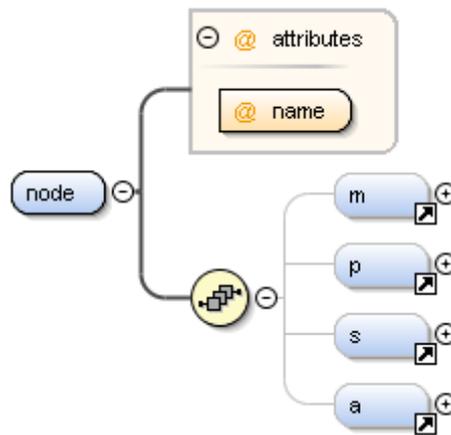


Abbildung 12: <node> Element

Ein <node> Element besitzt ein Attribut `name` welches den Namen der gewünschten Gruppierung aufweist. Jedes <node> Element kann dabei kein oder ein <p>, <m>, <s>, <a> Element aufweisen. Diese gruppieren innerhalb eines <node> Elements die Messungen, Parameter, Signale und Alarme welche es besitzt.

<p> Element

Dieses Element dient zur Gruppierung der Parameter welche auf dem besagten <node> Element möglich sind. Zum Beispiel um die maximale und minimale Position der Einspritzdüse zu setzen oder das minimale Niveau der automatischen Einschaltung einer Pumpe zu parametrieren.

<m> Element

Dieses Element dient zur Gruppierung der Messungen jeglicher Art welche für das entsprechende <node> Element möglich sind. Ein typisches Beispiel stellt dabei die Temperatur dar, welche gemessen beziehungsweise ermittelt wird und anschliessend im SCADA angezeigt wird.

<s> Element

Dieses Element dient zur Gruppierung der Signale welche auf dem besagten <node> Element möglich sind. Ein typisches Beispiel ist, der Status einer Pumpe: EIN / AUS.

<a> Element

Dieses Element dient zur Gruppierung der Alarme welche auf dem besagten <node> Element möglich sind. Ein typisches Beispiel ist, wenn der Notabschaltungsknopf der Turbine betätigt wurde oder falls der Geschwindigkeitssensor der Turbinenwelle einen Defekt besitzt beziehungsweise eine Störung aufweist.

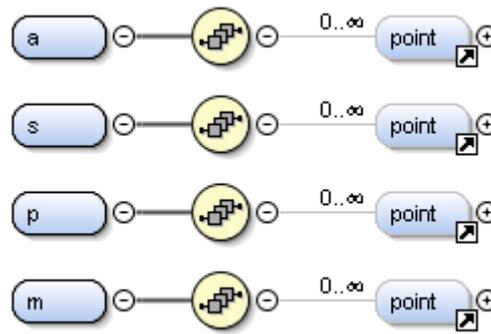


Abbildung 13: <a> <s> <p> <m> Elemente

Jedes dieser Elemente kann keine oder mehrere <point> Elemente besitzen.

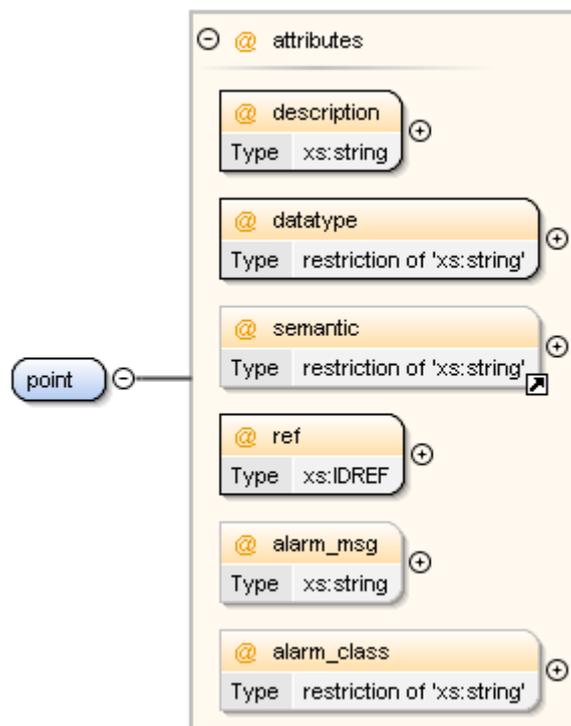


Abbildung 14: <point> Element

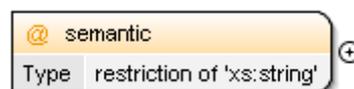


Abbildung 15: semantic Attribut

Ein <point> Element besitzt mehrere Attribute.

Das Attribut **description** kann als Beschreibung des entsprechenden Datentyps verwendet werden.

Das Attribut `datatype` besitzt eine Aufzählung welche zur Bestimmung des in der Supervisionssoftware verwendeten Datentyps dient. Folgende Einträge besitzt dieses Attribut:

- DINT
- INT
- REAL
- SINT
- UDINT
- BOOL
- BYTE
- WORD
- DWORD
- STRING
- STRING_8
- STRING_20
- STRING_40
- STRING_80

Das Attribut `semantic` besitzt eine Aufzählung welche weiter unten ebenfalls für das `<item>` Element (Im Abschnitt FIELD erklärt) verwendet wird. Dieses Attribut dient als Präsortierung, das heisst es beschreibt um welchen Typ von `<point>` es sich handelt. Ein `<point>` Element innerhalb eines `<m>` Elements beschreibt eine Messung dabei kann es sich um alle möglichen Arten von Messungen handeln (Temperatur, Geschwindigkeit, Vibration usw.) anhand des `semantic` Attributs kann einem `<point>` Element eine genauere Bedeutung zugeordnet werden.

Das Attribut `ref` dient als Verbindung (mapping) zwischen einem `<item>` Element und einem `<point>` Element (Im Abschnitt FIELD erklärt). Dabei werden automatisch alle ID's welche in den `<item>` Elementen unter dem Attribut `id` zugewiesen wurden angezeigt. Somit kann ein `<point>` Element mit einem beliebigen `<item>` Element verbunden werden. Dabei können diese Verbindungen unabhängig des `<device>` Elements durchgeführt werden das heisst es können verschiedene Prozessvariablen in einem `<node>` Element gruppiert werden unabhängig des sich darunter befindenden Gerätetyps.

Das Attribut `alarm_msg` kann verwendet werden um dem `<point>` Element mit dem Attribut `type` auf „alarm“ eine Alarm Nachricht zuzuweisen.

Das Attribut `alarm_class` kann verwendet um einem `<point>` Element eine Alarmpriorität zuzuweisen. Eine `alarm_class` Attribut kann nur 4 Buchstaben besitzen.

IEC61850

Die Norm IEC 61850 [6] ist eine Norm die von International Electrotechnical Commission (IEC) erstellt wurde und vor allem Anwendung im Bereich von Schutz- und Leittechnik Schaltanlagen von Mittel- und Hochspannungstechnik findet. Sie beschreibt wie Komponenten eines Kraftwerks beziehungsweise Wasserkraftwerks auszusehen haben und welche Daten bestimmte Komponenten davon besitzen. Die Struktur der Norm lässt sich vereinfacht in einem Beispiel aufzeigen.

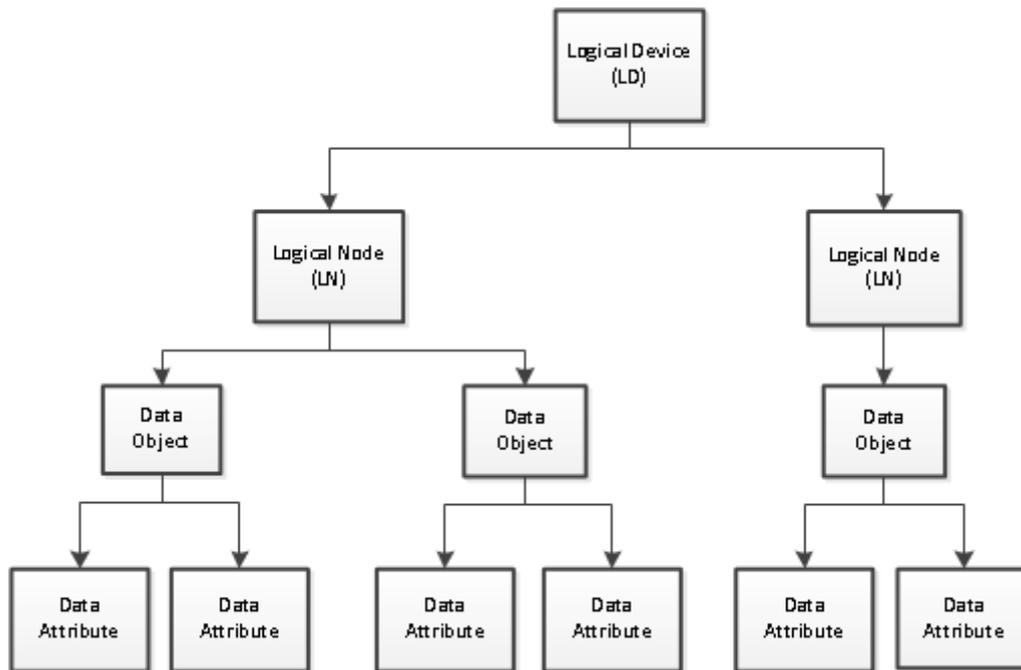


Abbildung 16: Prinzip der IEC61850 Norm

Die Norm IEC61850-7-410 zeigt eine Vielzahl von logical nodes (Klassen) und wie deren Datenstruktur auszusehen hat. Jede der Klassen besitzt optionale und Pflicht Daten Objekte. Jeder logical node kann unter einer logical device (Klasse) gruppiert werden. Die Klasse welche einen Deflektor einer Turbine laut IEC61850 Norm modelliert sieht wie folgt aus:

5.6.5 LN: Deflector control

Name: HDFL

Logical node HDFL shall be used to represent the deflector control of a Pelton turbine.

HDFL class				
Data Object Name	Common Data Class	Explanation	T	M/O
LNNName		Shall be inherited from Logical-Node Class (see IEC 61850-7-2)		
Data Objects				
<i>Common Logical Node Information</i>				
LocKey	SPS	Local or remote key		O
LocSta	SPC	Remote control blocked		O
OpCnt	INS	Operation counter		O
Loc	SPS	Local operation selected		M
<i>Status information</i>				
ClsPos	SPS	Closed end position reached (deflector cannot move further)		C ¹
OpnPos	SPS	Open end position reached (deflector cannot move further)		C ¹
Mvm	SPS	Deflector is moving		O
Stuck	SPS	Deflector is blocked (cannot move from present position)		O
DflMan	SPS	Manual operation of deflector is active		O
<i>Settings</i>				
OpnLim	ASG	Opening limit of deflector position (temporary restriction)		O
ClsLim	ASG	Closing limit of deflector position (temporary restriction)		O
Incr	ASG	Increment of position change for open / close commands		O
<i>Measured values</i>				
PosPc	MV	Deflector position given as 0 – 100 %		C ²
PosDeg	MV	Deflector position given as 0 – 90 °		C ²
Flw	MV	Calculated liquid flow through the deflector [m ³ / s]		O
<i>Controls</i>				
PosSpt	APC	Deflector position set-point		O
Opn	SPC	Deflector to full open position		O
Cls	SPC	Deflector to full closed position		O
PosChg	INC	Change deflector position (stop, raise, lower)		C ²
PosChgIncr	INC	Incremental change of position		C ²
BlkOpn	SPC	Block opening of the deflector		O
BlkCls	SPC	Block closing of the deflector		O

Note: For data attributes with conditions C¹, one or both may be used, however the use of at least one is mandatory. Data attributes with conditions C² are optional, but if used, only one can be selected.

Abbildung 17: Deflektor laut IEC61850-7-410 Norm

Es wurde versucht aus einer exportierten Liste eines SCADA Systems von HYDRO-Exploitation deren Einträge und Prozessvariablen allen IEC61850 bekannten logical nodes und deren Daten Objekten zuzuweisen. Für die oben gezeigte Klasse einer Einspritzdüse entspräche dies für das Daten Objekt **PosPc** unter **Measured values** dem von HYDRO-Exploitation im Anhang 4: Gr3 Liste superviseur.xls auffindbaren Eintrag:

Mesure	Adresse	ADD SCA	Libelle	Designation
42	%MW817	400817	PRSP_POS_LIMITEUR	Position limiteur électrique

Abbildung 18: Eintrag von HYDRO-Exploitation (Anhang 4: Gr3 Liste superviseur.xls)

Es ist jedoch nicht möglich alle Einträge von einem HYDRO-Exploitation Projekt einer IEC61850 Klasse und deren optionalen oder Pflicht Daten Objekten zuzuweisen. Der Grund liegt darin, dass für gewisse Einträge zusätzliche neue Klassen hinzugefügt werden müssten, welche das entsprechende Daten Objekt besitzen. Die Neu hinzugefügte Klasse hätte jedoch Pflicht Daten Objekte (durch M in der rechten Spalte der Abbildung 17: Deflektor laut IEC61850-7-410 Norm gekennzeichnet) welche nicht einem Eintrag aus der Liste von HYDRO-Exploitation zugeordnet werden können. Das heisst für gewisse Einträge würden zusätzliche Klassen hinzugefügt welche unter den benötigten Pflichtfeldern keine Einträge stehen hätten. Dies ist laut IEC61850 Norm nicht zulässig.

Um das IEC61850 Protokoll in dem framework.xml Dokument zu implementieren könnten die von der Norm verwendeten XML Dokumente eines IEC61850 kompatiblen Geräts (z.B. Turbine, Deflektor, Pumpe usw.) oder des nicht verwendeten IEC61850 Schutzgeräts verwendet werden um den SCADA Abschnitt automatisch mit den benötigten Informationen aufzufüllen.

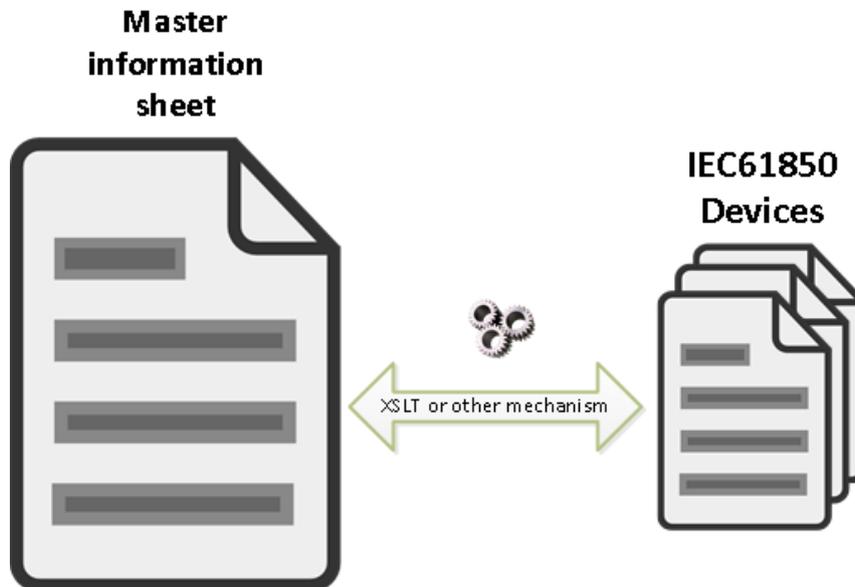


Abbildung 19: Einbindung von IEC61850 Geräten

Dieser Abschnitt wurde im Rahmen der Diplomarbeit nicht umgesetzt und soll in einem künftigen Projekt integriert werden.

FIELD

Der Abschnitt FIELD ist eine Eigenentwicklung, welche sich auf der Analyse der beiden im Kapitel SCADA Entwicklungsprozess befindenden Architekturen und deren verwendeten Programmen stützt. Alle benötigten Informationen die von den verschiedenen Programmen und Protokollen benötigt werden, wurden evaluiert. Folgende Architektur dient dabei als Grundgerüst:

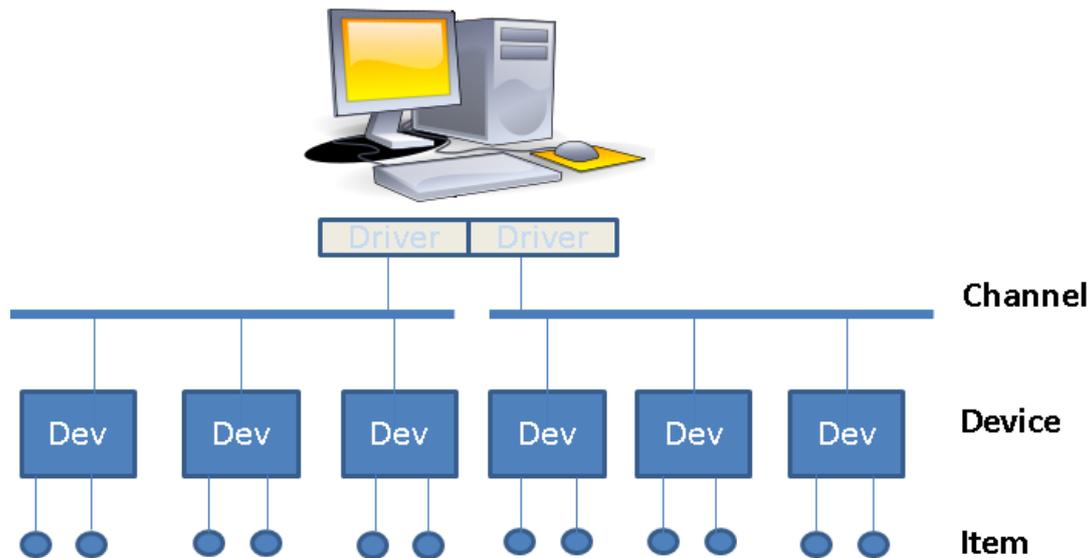


Abbildung 20: Architektur für FIELD Abschnitt

Es wurde versucht eine ähnliche Architektur aufzubauen, wobei ein Item einen physikalischen Ein / Ausgang auf der SPS modelliert. Ein Device modelliert ein Gerät zum Beispiel ein Quantum, M340 oder S7-300. Der Channel modelliert ein Feldbus-System mit dem sich darin verwendetem Protokoll. Diese Struktur soll gültig für eine Architektur mit OPC wie auch ohne sein.

UML Design

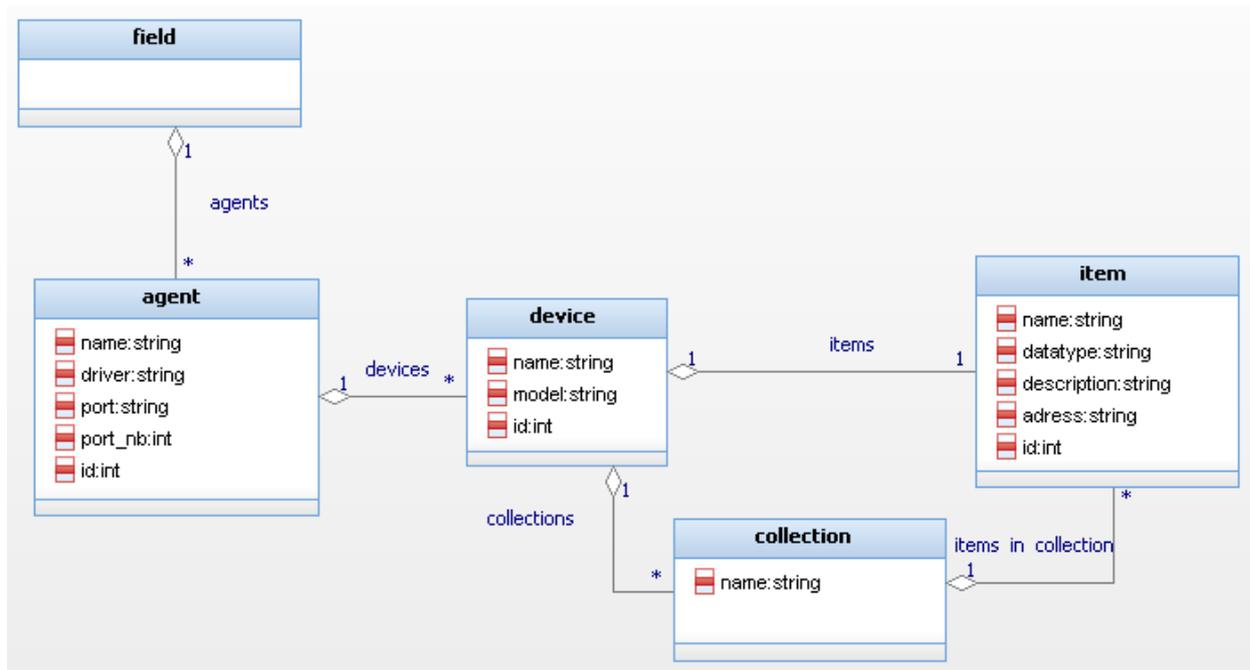


Abbildung 21: UML FIELD Architektur

Das UML Diagramm kann als Model der in der Abbildung 1: Architektur ohne OPC oder Abbildung 2: Architektur mit OPC Architektur verstanden werden. Unabhängig der gewählten Architektur trifft das Diagramm zu und kann für beide Architekturen verwendet werden.

XSD Design

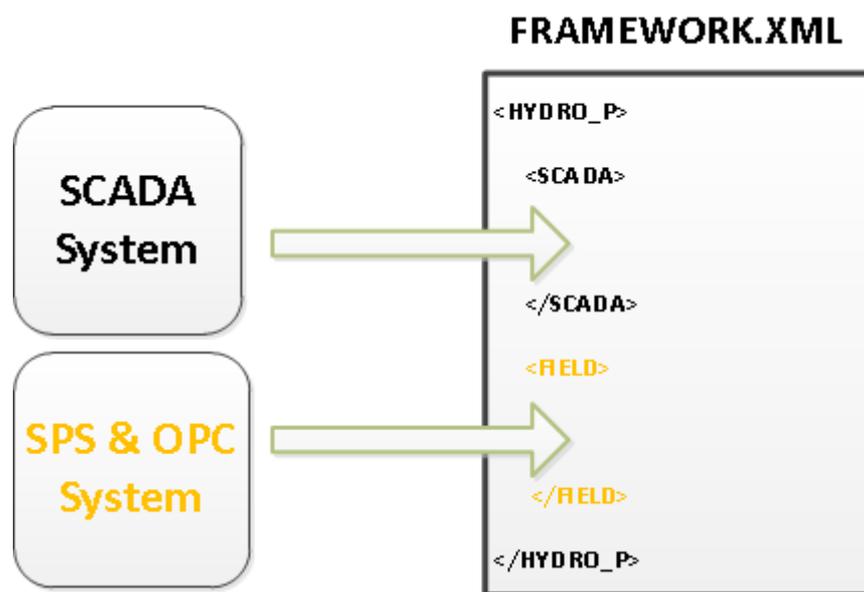


Abbildung 22: FIELD Abschnitt des framework.xml Dokuments

Das in der Abbildung 21: UML FIELD Architektur gezeigte UML Diagramm kann in ein XML Schema (XSD) übersetzt werden. Das daraus resultierende Schema dient als Grundgerüst für jedes neue oder bestehende Projekt welches die Geräte, Prozessvariablen und Protokolle der Wasserkraftwerk-Infrastruktur darstellen soll. Ein vereinfachtes XML Beispiel erleichtert das Verständnis des folgenden Kapitels:

```
<field>
  <agent id="1" name="NameOfField" driver="Driver" port="Port">
    <device id="11" name="NameOfSPS" devicetype="Type">
      <address>
        <modbus/>
        <ip address="153.109.5.240"/>
      </address>
      <items>
        <collection name="NameOfCollection">
          <item id="ID-000001" name="Input_2" datatype="Boolean" rights="rights" >
            <description>comment</description>
            <address>I0.2</address>
          </item>
        </collection>
      </items>
    </device>
  </agent>
</field>
```

Aufbau

Die Richtlinie für den FIELD Abschnitt im XML Schema, sieht wie folgt aus und kann im Anhang 3: framework.xsd betrachtet werden.

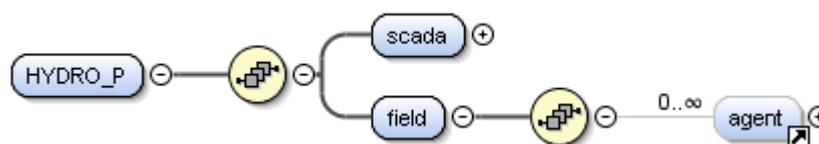


Abbildung 23: <field> Element

Ein Projekt besitzt ein <field> Element. Ein <field> Element kann 0 bis unendlich viele <agent> Elemente besitzen.

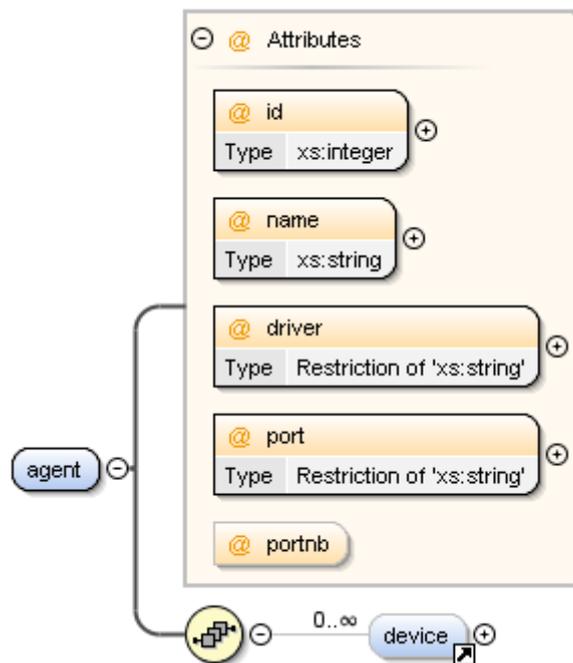


Abbildung 24: <agent> Element

Ein <agent> Element besitzt mehrere Attribute und kann kein oder unendlich viele <device> Elemente besitzen.

Das Attribut **id** ist eine Zahl des Typs Integer und dient als Identifikation des <agent> Elements. Jedes dieser Elemente sollte einen einzigartigen Wert innerhalb der Projekts besitzen. Bemerkung: die Einzigartigkeit dieses Elements wird nicht von XML überprüft! Es können durchaus doppelte **id** Werte vorhanden sein.

Das Attribut **name** dient als Name des <agent> Elements und dient als Bezeichnung des physikalisch angeschlossenen Feldes.

Das Attribut **driver** besitzt eine Aufzählung verschiedener zurzeit implementierter Treiber auf dem Kepware Server.

- Modbus Ethernet
- Modbus Serial
- Siemens TCP/IP Ethernet

Das Attribut **port** besitzt ebenfalls eine Aufzählung der verschiedenen zulässigen physikalischen Anschlüsse auf welchem das Feldsystem angeschlossen ist:

- Ethernet
- Serial

Diese Parameter sind hauptsächlich für den Unterschied zwischen den Modbus Systemen gedacht.

Das Attribut **portnb** dient als Erweiterung des vorherigen Attributs **port**. Somit kann einem seriellen Port sein Nummer zugewiesen werden z.B **port**="Serial" **portnb**="1" entspricht dem COM 1 Port.

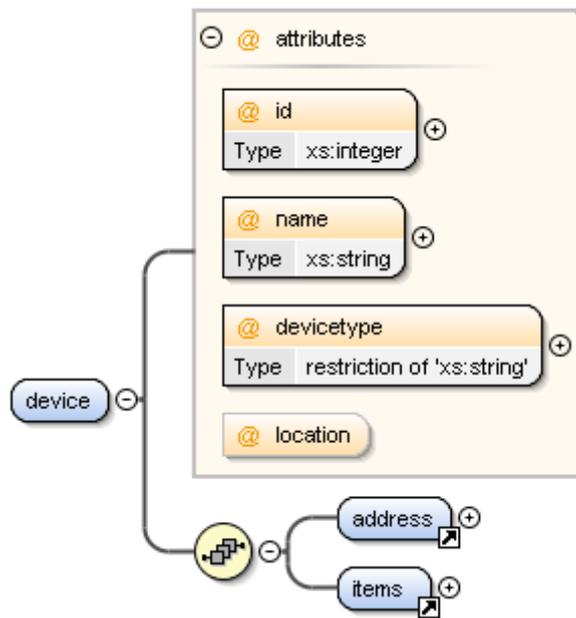


Abbildung 25: <device> Element

Ein <device> Element besitzt mehrere Attribute des Weiteren besitzt jedes <device> Element ein <address> Element und ein <items> Element.

Das Attribut **id** ist eine Zahl des Typs Integer und dient als Identifikation des <device> Elements jedes dieser Elemente sollte einen einzigartigen Wert innerhalb der Projekts besitzen. Bemerkung: die Einzigartigkeit dieses Elements wird nicht von XML überprüft! Es können durchaus doppelte **id** Werte vorhanden sein.

Das Attribut **name** wird als Name des <device> Elements benutzt und dient als Bezeichnung der physikalischen SPS.

Das Attribut **devicetype** besitzt eine Aufzählung welche alle Modelltypen gruppiert die vom Kepware unterstützt werden. Folgende Modelle von SPS werden zurzeit implementiert:

- Siemens:
- S7-200
- S7-300
- S7-400
- S7-100
- Modbus RTU Device (Quantum oder sonstige Modbus RTU Protokoll benutzende Geräte)
- Modbus Eth Device (Quantum oder sonstige Modbus Ethernet Protokoll benutzende Geräte)

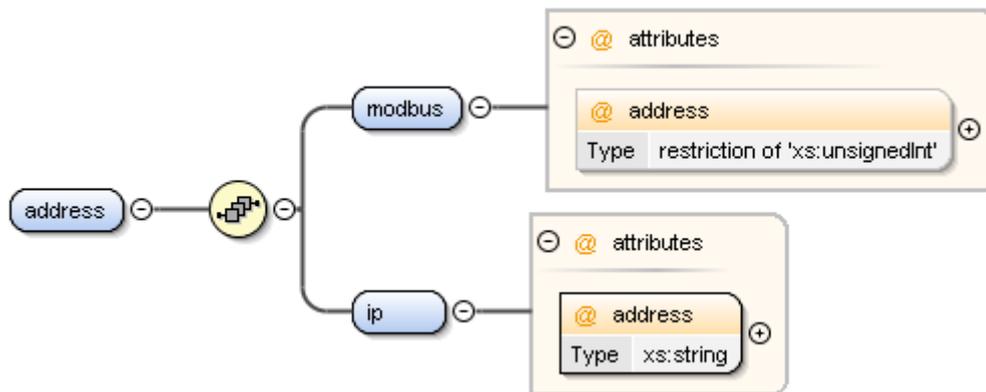


Abbildung 26: <address> Element

Ein <address> Element besitzt ein <modbus> Element und ein <ip> Element. Das <address> Element dient zur Bestimmung der Adresse eines <device> Elements. Das <modbus> Element besitzt ein Attribut namens **address**. Dieses kann verwendet werden um einem <device> seine Modbusadresse anzugeben. Das <ip> Element besitzt ebenfalls ein **address** Attribut, in welchem die IP-Adresse des angeschlossenen Geräts angegeben werden kann.

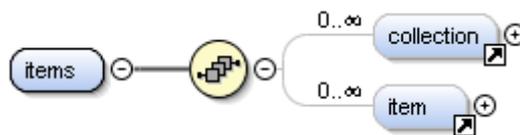


Abbildung 27: <items> Element

Ein <items> Element besitzt keine oder unendlich viele <collection> und <item> Elemente

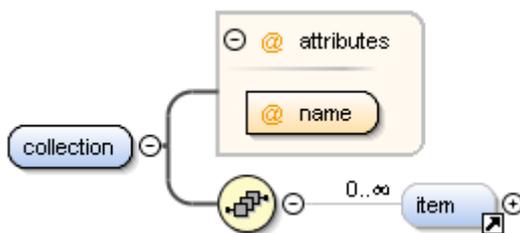


Abbildung 28: <collection> Element

Ein <collection> Element hat kein oder unendlich viele <item> Elemente. Das <collection> Element hat ein Attribut namens **name** welches einem Ordernamen der enthaltenen <item> Elemente entspricht.

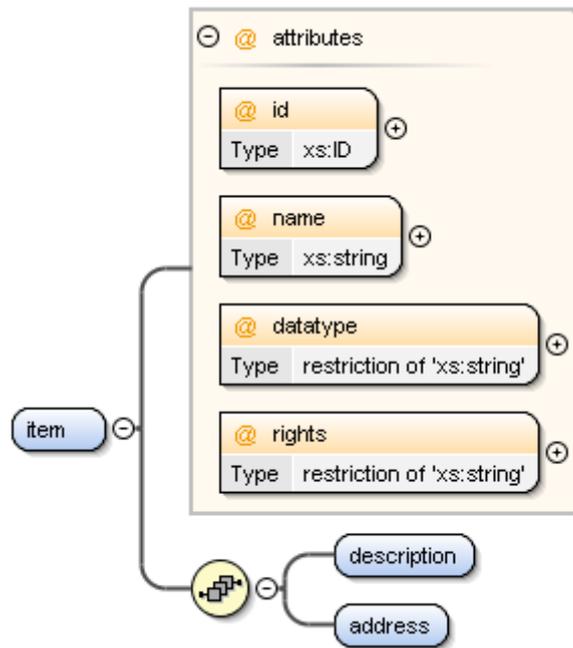


Abbildung 29: <item> Element

Ein <item> Element besitzt mehrere Attribute, ein <description> und ein <address> Element.

Das Attribut **id** dient als einzigartige Identifikation eines <item> Elements innerhalb des Dokuments. Bemerkung: doppelte Einträge werden vom Dokument geprüft und als fehlerhaft markiert.

Das Attribut **name** wird verwendet um dem <item> Element einen Namen zuzuordnen.

Das Attribut **datatype** besitzt eine Aufzählung welche dem <item> Element seinen Datentyp zuweist. Alle möglichen Datentypen sind:

- Boolean
- Char
- Byte
- Short
- Word
- Long
- DWord
- Float
- Double
- String
- BCD
- LBCD
- Date
- Boolean Array
- Char Array
- Byte Array

- Short Array
- Long Array
- DWord Array
- Float Array
- Double Array
- String Array
- BCD Array
- LBCD Array
- Date Array

Das Attribut **rights** besitzt eine Aufzählung, welche die Zugriffsrechte auf einem `<item>` Element beschreiben. Folgende Rechte sind möglich:

- Read Only
- Read/Write

Das `<description>` Element wird als Beschreibung des `<item>` Elements verwendet.

Das `<address>` Element beinhaltet die Adresse des Ein- oder Ausgangs auf dem SPS. Bemerkung: Diese wird vom XML Schema nicht auf ihre Richtigkeit überprüft. Das bedeutet es können irgendwelche Zeichen eingegeben werden. Es obliegt dem Systemintegrierer die Richtigkeit der Adressen zu bestimmen.

Beispiel einer erstellten Infrastruktur

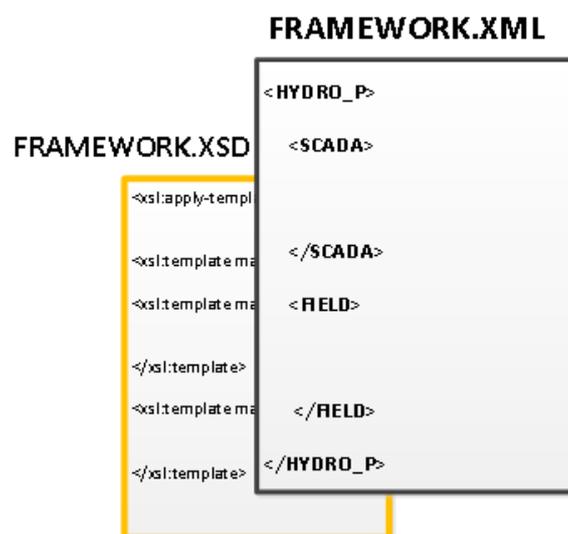


Abbildung 30: framework.xml basierend auf framework.xsd

Im Anhang 5: framework.xml kann ein Beispiel ein auf der XSD basierendes XML Dokument betrachtet werden. Dieses entspricht der entwickelten Richtlinie framework.xsd. Es können ohne

weiteres neue Elemente im SCADA und FIELD Abschnitt hinzugefügt werden, solange die vorgegebene Richtlinie eingehalten wird.

Bemerkung:

Folgender Code im Anfangsbereich des Dokuments muss manuell hinzugefügt werden und wird nicht von dem XML Schema kontrolliert.

```
<!DOCTYPE HYDRO_P [  
  <!ATTLIST item  
    id ID #REQUIRED  
  >  
>
```

Dieser Code bewirkt, dass das Attribut id eines `<item>` Elements dem Typ ID entspricht und somit die automatische Referenz in einem `<point>` Element korrekt funktioniert. Dieser Code entspricht einem DTD Code und ist mit einem XML Schema Code zu vergleichen.

Bemerkung:

Unter dem `<address>` Element eines `<item>` Elements welches sich innerhalb eines Modbus Geräts befindet, befindet sich ein Eintrag der folgendermassen aussieht:

```
<item id="ID-000025" name="PRSP_PRES_DEFL" datatype="Word" x:semantic="pressure"  
rights="Read/Write">  
  <description>Rack 3 slot 1</description>  
  <address>MW813</address>  
</item>  
<item id="ID-000019" name="PRSP_TH_LIM_MEC" datatype="Boolean" x:semantic="temperature"  
rights="Read Only">  
  <description>Rack 1 slot 5</description>  
  <address>I1.5.9 => 100009</address>  
</item>
```

Man erkennt, dass die Adresse des ersten `<item>` Elements als MW813 beschrieben wird. Dies entspricht einer internen Speicherposition einer SPS welche mittels Unity V7 programmiert wird.

Da der KEPServerEXV5 nur reine Modbusadressen (400xxx, 300xxx, 100xx und 000xxx) erkennt, muss die Adresse MW813 in eine Modbusadresse konvertiert werden. Dieser Konversionsprozess wird automatisch durchgeführt.

Die Adresse des zweiten `<item>` Elements besitzt ein Trennzeichen => welches auf der linken Seite einen Ein/Ausgang in Unity V7 beschreibt und auf der rechten Seite eine Modbusadresse. Durch das erwähnte Trennzeichen wird in der Unity V7 Software die linke Seite verwendet um die Prozessvariable zu adressieren. Die rechte Seite des Trennzeichens wird in der KEPServerEXV5 Software verwendet um dieselbe Prozessvariable zu adressieren. Einerseits ist die Prozessvariable auf der SPS durch I1.5.9 definiert und andererseits in der KEPServerEXV5 Software durch 100009 was

grundsätzlich dasselbe ist. Für das Prinzip der Konversion der Adressen siehe Kapitel: KEPServerEXV5 und Unity V7.

Diese Vorgehensweise Impliziert jedoch, dass in der SPS der entsprechende Ein/Ausgang auf dem korrekten internen Speicher (Modbusadresse) referenziert wird.

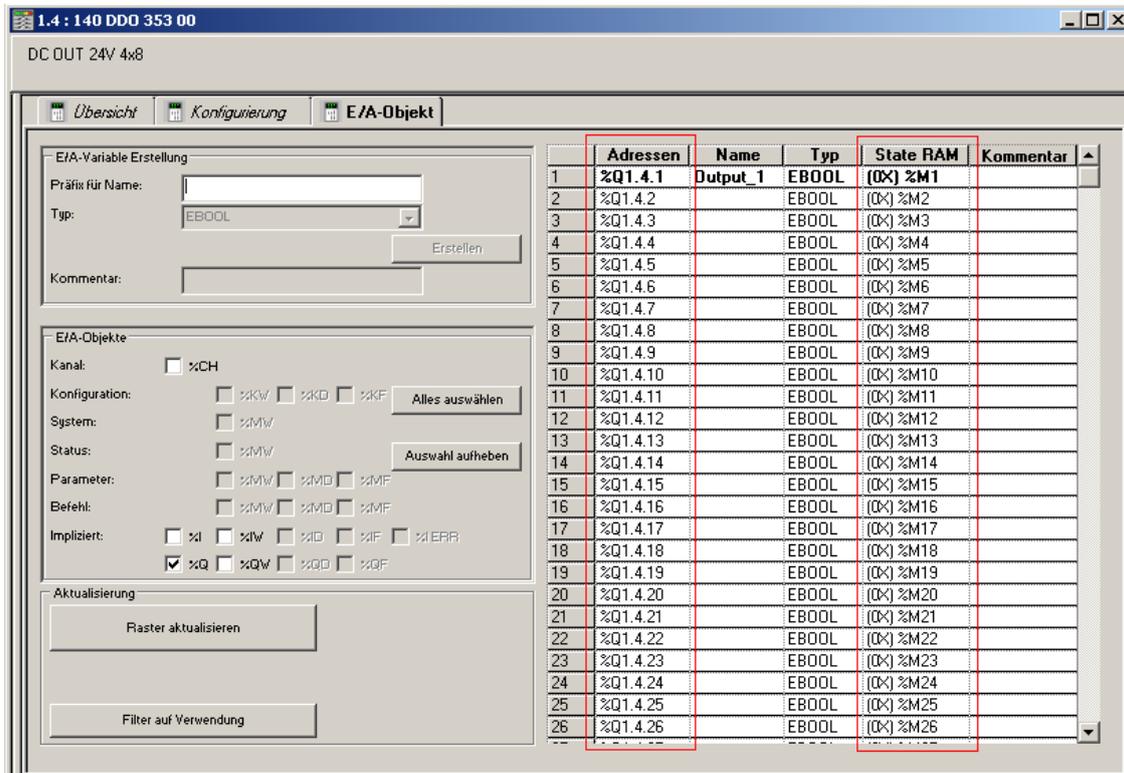


Abbildung 31: Digitale Ausgangskarte in Unity V7

Beispiel einer Digitalen Ausgangskarte und der Verbindung zwischen den Ein- und Ausgängen und dem internen Speicher und die daraus resultierenden Modbusadressen.

Referenzumgebung & dazugehörige XSLT

oXygene V12

oXygene V12 ist ein XML Editor der verwendet wurde um alle XML, XSD und XSLT Dokumente zu erstellen. Dieser Editor erlaubt es bei den XSLT zwischen verschiedenen XSLT Prozessoren zu wählen. Grundsätzlich wurde für die XSL Transformationen der Saxon-HE 9.4.0.6 Transformator verwendet. Alle XSLT wurden in der XSLT Version 2.0 geschrieben um bestimmte Funktionen und Methoden nutzen zu können.

Cimplicity

Cimplicity ist ein SCADA Programm welches verschiedene Protokolle unterstützt und implementiert. Es kann vielseitig eingesetzt werden und wird von HYDRO-Exploitation verwendet um ihre Infrastrukturen darzustellen. Cimplicity kann als ein OPC Client mit einer Anbindung an mehrere OPC Server funktionieren, wobei der OPC Server alle Prozessvariablen besitzt und der Client darauf zugreifen kann.

Cimplicity benötigt ein .csv Dokument um dessen Prozessvariablen, in diesem Fall Point's genannt, zu importieren. Dieses .csv Dokument besitzt ein Komma als Trennzeichen. Das gewünschte Format eines zu Importierenden Dokuments besitzt 102 Kolonnen.

PT_ID	ACCESS	ACCESS_FILTER	ACK_TIMEOUT	ADDR	...	UPDATE_CRITERIA	VARIANCE_VAL	VAR5
##								
MTU.TURB.m.PRSP_VIT_ROT_ARB	R			\$(MTU)SF.PLC1.MTU.PRSP_VIT_ROT_ARB	...	UC		1
MTU.TURB.s.PRSP_MIP_VIT_95	R		-1	\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_95	...	UC		1
MTU.TURB.s.PRSP_MIP_VIT_80	R		-1	\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_80	...	UC		1
MTU.TURB.a.PRSP_MIP_VIT_120	R		-1	\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_120	...	UC		1
MTU.DEFL.p.PRSP_MIP_VIT_10	R			\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_10	...	UC		1
MTU.DEFL.p.PRSP_MIP_VIT_12	R			\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_12	...	UC		1
MTU.DEFL.s.PRSP_MIP_VIT_11	R		-1	\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_11	...	UC		1
MTU.DEFL.s.PRSP_MIP_VIT_13	R		-1	\$(MTU)SF.PLC1.MTU.PRSP_MIP_VIT_13	...	UC		1

Abbildung 32: Format eines zu Importierenden .csv Dokuments für Cimplicity

Jede dieser Kolonnen hat einen bestimmten Einfluss auf die Konfiguration der Prozessvariablen. Ein vollständiges .xlsx Dokument welches von HYDRO-Exploitation verwendet wird kann im Anhang 2: LIST Import.xlsx betrachtet werden. Einige Kolonnen wurden dort ausgeblendet.

Aus dem Dokument **LIST Import.xlsx** und der Hilfe von Cimplicity wurde eine XSLT erstellt die alle benötigten Informationen umfasst um eine Infrastruktur in Cimplicity zu laden. Dabei wurden die Points in einer logischen Ordner Struktur unterteilt. Um einen Ordner zu erstellen wird unter der Kolonne **PT_ID** ein Punkt verwendet. Das Prinzip entspricht dabei dem Muster `<foldername>.<foldername>.<foldername>.<pointname>` wobei der letzte Name den eigentlichen Point Namen beinhaltet. Um einen Point einer OPC Gruppe zuzuweisen wird in der Kolonne **ADDR** folgende Syntax verwendet:

`$(GROUPNAME) <path of point in KEPServerEXV5>`

GROUPNAME enthält den Namen der OPC Gruppe welche vom Client aus auf dem Server erstellt wird (Prinzip von OPC kann im Anhang 1: Bericht OPC unter 2.3.3 IOPCServer gefunden werden). Die Charakteristiken der OPC Gruppen können im .ini Dokument des entsprechenden Ports erstellt und parametrisiert werden (siehe Kapitel: XSLT OPC Groups).

XSLT Cimplicity

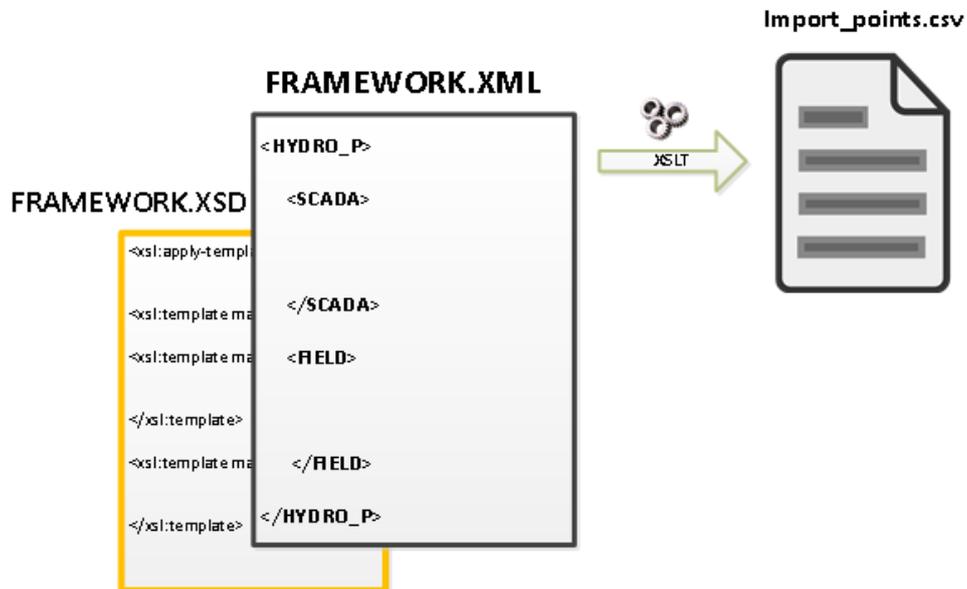


Abbildung 33: XSLT zur Generierung des import_points.csv Dokuments

Die XSL Transformation, welche aus dem framework.xml Dokument das gewünschte für Cimplicity zu importierende import_points.csv Dokument erstellt, kann im Anhang 6: XSLTcimplicity.xml gefunden werden.

Um ein Ausgabe Dokument zu erstellen wird die `<xsl:result-document href="{Cimplicity}"></xsl:result-document>` Funktion verwendet. Diese benötigt eine Variable `Cimplicity` welche den Namen des zu erstellenden Dokuments und dessen Pfad beinhaltet.

Dieses Dokument befindet sich nach Ausführung der Transformation im Verzeichnis Cimplicity/Points da dies der Wert der `Cimplicity` Variable ist.

Die Namen der Kolonen werden durch das `<xsl:text></xsl:text>` Element gekennzeichnet und untereinander mit dem Kommatrennzeichen getrennt.

Die Schablonen-Regel `<xsl:apply-templates select="//fr:point"/>` erstellt für jedes `<point>` Element im framework.xml Dokument eine Zeile mit den entsprechenden 102 Informationen. Einige Informationen sind statisch, kommen daher aus dem XSLTcimplicity.xsd Dokument, andere werden aus dem framework.xml Dokument dynamisch eingesetzt. Elemente werden mit dem `fr:` Präfix bezeichnet. Der `fr:` Präfix bezeichnet einen Namensraum damit die XSLT weiss aus welcher Umgebung und welchem Raum das entsprechende Element kommt. Der Namensraum wird im `<xsl:stylesheet>` definiert.

Zum Beispiel wird die **PT_ID** Kolone wie folgt aufgebaut:

```
<!-- PT_ID -->
<xsl:value-of select="ancestor::fr:group/@name"/><xsl:text>.</xsl:text>
<xsl:value-of select="ancestor::fr:node/@name"/><xsl:text>.</xsl:text>
<xsl:value-of select="name(..)"/><xsl:text>.</xsl:text>
<xsl:value-of select="id(@ref)/@name"/>
<xsl:text>,</xsl:text>
```

Mittels `ancestor::` Befehl kann vom jetzigen Element aus (in diesem Fall `<point>`, da man mit `fr:point` sich zurzeit auf einem `<point>` Element befindet) auf jedem beliebigen übergeordneten Element navigiert werden.

Mittels `name(..)` kann vom jetzigen Element aus der Name des übergeordneten Elements erhalten werden. Was laut Struktur des framework.xml Dokuments in diesem Fall einem `<m>`, `<p>`, `<s>` oder `<a>` Element entsprechen würde .

Mittels `id(@ref)` kann vom jetzigen Element aus, welches eine Referenz zu dem Originalen Element besitzt, navigiert werden.

Ein `<point>` Element, welches sich innerhalb eines `<m>` oder `<p>` Elements befindet (Messung oder Parameter), sieht identisch in seinen Kolonnen aus. Ein `<point>` Element welches sich innerhalb eines `<s>` oder `<a>` Elements befindet (Signalisation oder Alarm) sieht identisch in seinen Kolonnen aus unterscheidet sich jedoch von den Messungen und Parameter `<point>` Elementen.

Eine solcher Unterschied wird mit einer Bedingung `<xsl:if test="name(..)='a' or name(..)='s'">` geregelt. Falls der Name des übergeordneten `<point>` Elements ein a oder ein s ist (Alarm oder Signalisation) wird ein Parameter der Kolonne hinzugefügt ansonsten nicht, da es sich nicht um eine Messung oder einen Parameter handelt.

Ein Beispiel ist die Kolonne **ACK_TIMEOUT**

```
<!-- ACK_TIMEOUT -->
<xsl:if test="name(..)='a' or name(..)='s'">
<xsl:text>-1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
```

Am Ende jedes `<point>` Elements wird eine neue Linie angefangen. Diese wird mittels `<xsl:text>
</xsl:text>` gekennzeichnet. `` steht für carriage return und `
` für new line.

Bemerkung:

- Jedes `<group>` Element entspricht in Cimplicity einer Resource (name des `<group>` Elements)
- Jedes `alarm_class` Attribut entspricht in Cimplicity einer Alarm Klasse.
- Jedes `<group>` Element entspricht in Cimplicity einer OPC Gruppe.

KEPServerEXV5

KEPServerEXV5 ist ein Multiprotokoll OPC Server der vom Hersteller Kepware produziert wird. Dieser beinhaltet dabei Mehrere Protokolle die im Rahmen des Projekts verwendet werden. Die verwendeten Plug-Ins sind Siemens OPC Server Suite und die Power Distribution Suite. Diese beinhalten alle Siemens Protokolle der S7 Familie, die Modbus TCP/IP, Modbus RTU Serial, Modbus ASCII Serial und IEC61850 MMS Client Protokolle.

Der KEPServerEXV5 OPC Server besitzt die Möglichkeit ein ganzes Projekt in einem XML-Format zu importieren. Unter Projekt versteht KEPServerEXV5 die Konfiguration des Channels, dessen angeschlossenen Geräten und deren dazugehörigen Tags.

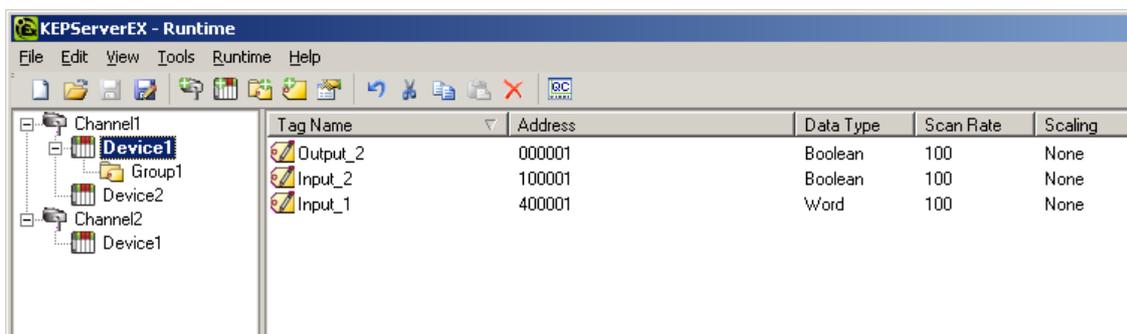


Abbildung 34: KEPServerEX V5 Design

Jeder Channel besitzt dabei verschiedene Parameter die konfiguriert werden können. Jeder dieser Parameter kann innerhalb des project.xml Projektdokuments gefunden werden. Dasselbe gilt für die Parameter eines Geräts und dessen Items. Die Struktur des zu erstellenden project.xml Dokuments unterliegt dabei einer Richtlinie die vom KEPServerEXV5 unter den Installationsdateien vorzufinden ist. Diese Richtlinie kann auch im Anhang 7: servermain.xsd Dokument betrachtet werden. Entspricht das project.xml Dokument nicht der servermain.xsd Richtlinie, kann das Projekt nicht in KEPServerEXV5 importiert werden. Fehlerhafte Importationen werden im KEPServerEXV5 im unteren Logbereich signalisiert.

XSLT Kewpare

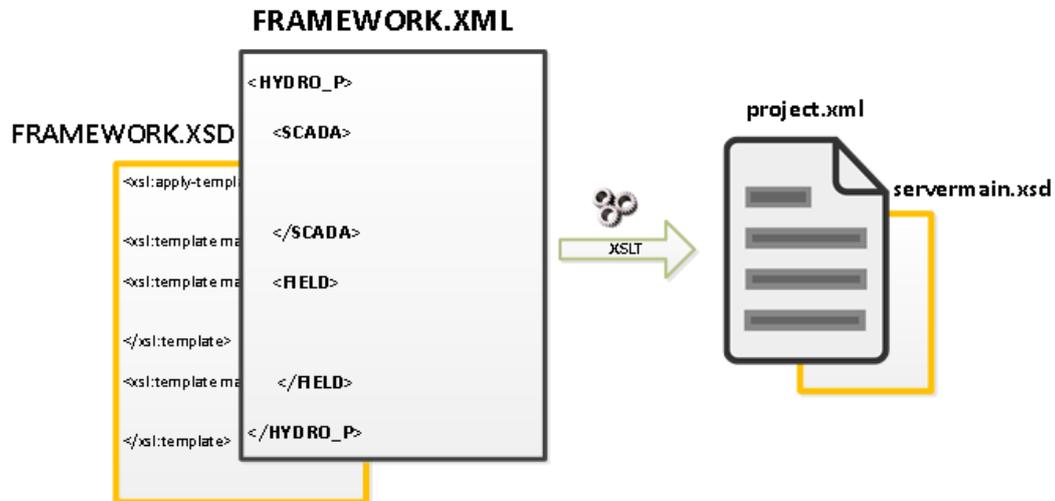


Abbildung 35: XSLT Kewpare Prinzip

Die XSLT, welche von KEPServerEXV5 benötigt wird und dazu dient ein project.xml Dokument zu generieren, besitzt einige statische Abschnitte. Diese erzeugen im project.xml Dokument statische Elemente, die von dem XSLTKewpare.xml Dokument erstellt werden. Einige Abschnitte der XSLT erzeugen wiederum gemischte Abschnitte in denen Informationen aus dem framework.xml Dokument und dem XSLTKewpare.xml Dokument verwendet werden. Vereinfacht kann die XSLT folgendermassen dargestellt werden:

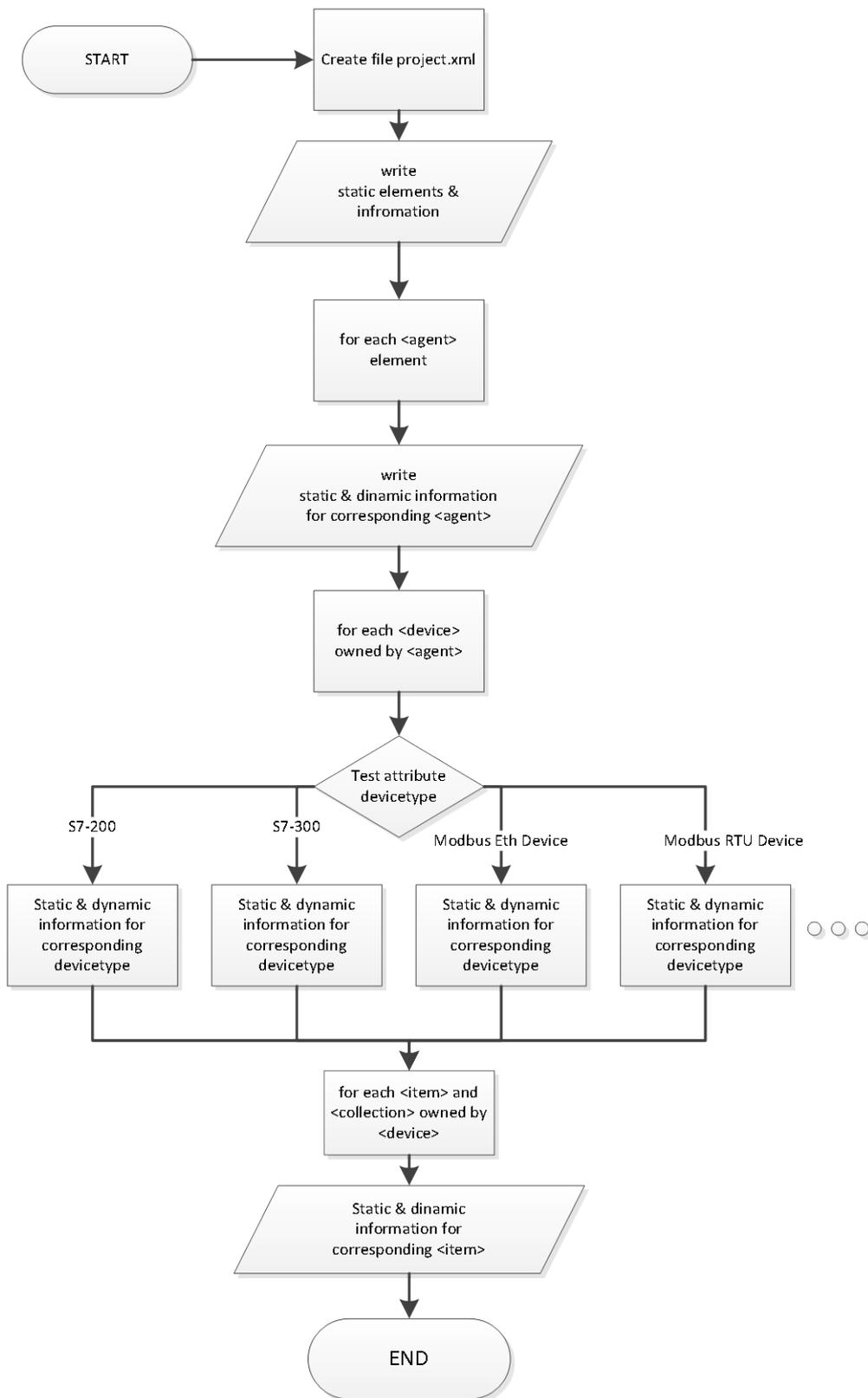


Abbildung 36: Flussdiagramm XSLT Kepware

Grundsätzlich werden ähnliche Mechanismen und Funktionen wie bereits im Kapitel Referenzumgebung & dazugehörige XSLT erklärt verwendet.

Das bereits angesprochene Problem der Adressierung einer Prozessvariable und deren Konversion in eine Modbusadresse wird für KEPServerEXV5 wie folgt durchgeführt:

```
<!-- if device type is a modbus -->
<xsl:if test="contains(ancestor::fr:device/@devicetype,'Modbus')">

  <!--
  if there is => in the address tag check if there is a memory address on the right side of =>
  if true then convert right side into modbusaddress
  else put right side of <address> tag into the address field of tag
  -->
  <xsl:choose>
    <xsl:when test="contains(fr:address,' => ')">
      <xsl:choose>
        <xsl:when test="contains(substring-after(fr:address,' => '), 'MW')"> [18 lines]
        <xsl:when test="contains(substring-after(fr:address,' => '), 'M')"> [18 lines]
        <xsl:when test="contains(substring-after(fr:address,' => '), 'IW')"> [18 lines]
        <xsl:when test="contains(substring-after(fr:address,' => '), 'I')"> [18 lines]
        <xsl:otherwise>
          <!-- take content right of => to put into address field of tag -->
          <xsl:value-of select="substring-after(fr:address,' => ')" />
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:choose>
        <xsl:when test="contains(fr:address,'MW')"> [18 lines]
        <xsl:when test="contains(fr:address,'M')"> [18 lines]
        <xsl:when test="contains(fr:address,'IW')"> [18 lines]
        <xsl:when test="contains(fr:address,'I')"> [18 lines]
        <xsl:otherwise>
          <!-- nothing invalid memory address-->
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:if>
<!-- if device type is a siemens -->
<xsl:if test="contains(ancestor::fr:device/@devicetype,'S7')">
  <xsl:value-of select="fr:address"/>
</xsl:if>
```

Abbildung 37: XSLT Code für die Adresskonvertierung

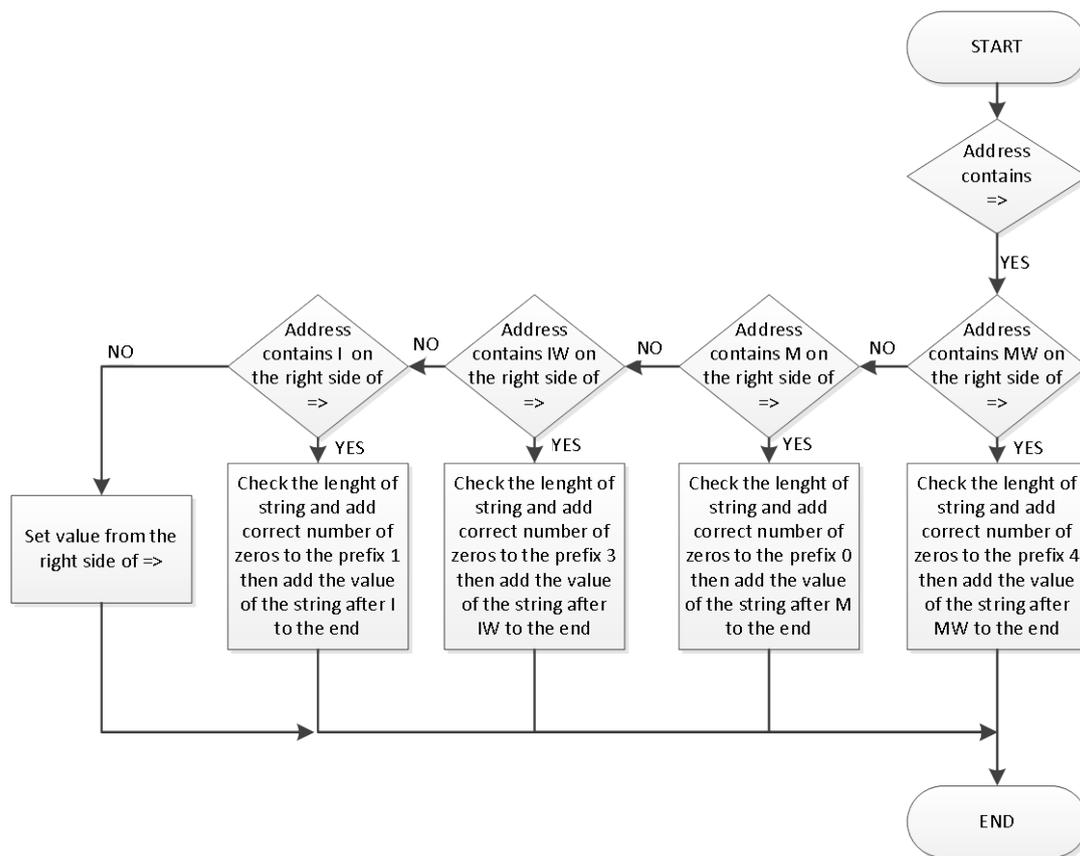


Abbildung 38: Konversion des <address> Elements mit Trennzeichen

Der Wert des <address> Elements eines <item> Elements welches unter einem Modbus Gerät steht kann verschieden aussehen. Falls das <address> Element ein => Trennzeichen besitzt wird für das project.xml Dokument die rechte Seite vom Trennzeichen untersucht. Handelt es sich dabei um eine interne Speicherposition wird diese automatisch in eine Modbusadresse konvertiert. Eine interne Speicherposition beginnt mit den Präfixen: **MW**, **M**, **IW** oder **I**. Falls es sich um eine dieser internen Speicherpositionen handelt, wird die Länge der anschliessenden Zahl kontrolliert. Da eine Modbusadresse immer gleich viele Zeichen besitzt nämlich 6 (gibt auch eine Version mit 5 Zeichen) kann zum Präfix **4,3,1** oder **0** die richtige Anzahl nullen hinzugefügt werden bevor der eigentliche Wert der Adresse hinzugefügt wird.

Beispiel:

<address>MW25</address>

- Besitzt MW somit beginnt der zu setzende Wert im project.xml Dokument mit 4 an.
- MW25 besitzt nach MW 2 Stellen, somit werden zur bereits gesetzten 4 drei weitere Nullen hinzugefügt: 4000
- Zuletzt wird die Zahl nach MW angefügt: 400025

Folgende Interne Speicherpositionen der Steuerung entsprechen einer Modbusadresse

I/O	Internal Memory	Modbus address	comment
%Qx.x.x	%Mxxx	000xxx	output coil
%Ix.x.x	%lxxx	100xxx	input coil
%IWx.x.x	%lWxxx	300xxx	input register
%QWx.x.x	%MWxxx	400xxx	output register

Abbildung 39: Interner Speicher referenzen Tabelle auf Modbusadresse

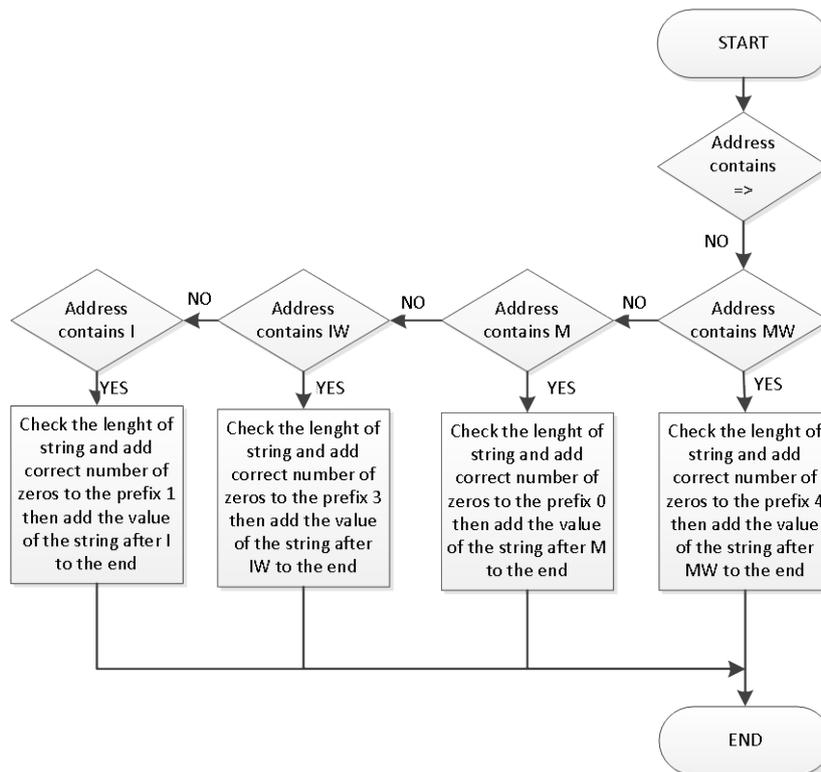


Abbildung 40: Konversion des <address> Elements ohne Trennzeichen

Falls sich auf der rechten Seite des Trennzeichens => nicht um eine interne Speicherposition handelt wird der sich darin befindende Wert in das project.xml Dokument an der Position der Tag Adresse hinzugefügt.

Eine weitere Möglichkeit ist, dass sich kein Trennzeichen innerhalb des <address> Elements befindet. Ist dies der Fall, wird untersucht ob es sich um eine reine interne Speicherposition handelt. Handelt es sich um eine interne Speicherposition, wird wieder eine Konversion in eine Modbusadresse vorgenommen. Falls nicht, wird keine Adresse dem Tag im project.xml hinzugefügt.

Der Code des XSLTKeppure.xsl Dokuments der Transformation, welches benötigt wird um die Konfiguration des KEPServerEXV5 zu generieren, kann im Anhang 8: XSLTKeppure.xsl Dokument betrachtet werden. Um eine neue oder bestehende Konfiguration in die KEPServerEX Software zu laden wird eine Batchdatei verwendet. Diese stoppt den KEPServerEXV5 Prozess und kopiert das erstellte project.xml Dokument in das C:\ProgramData\Kepware\KEPServerEX\V5 Verzeichnis. Anschliessend wird der KEPServerEXV5 Prozess wieder gestartet damit der Server project.xml das Dokument wieder laden kann. Achtung diese Art ein Projekt in KEPServerEX zu laden funktioniert nur

solange keine DCOM Konfigurationen vorgenommen wurden. Mehr Informationen dazu finden sich in der KEPServerEX Hilfe unter dem Import Kapitel.

Bemerkung:

Alle Channels und deren Devices werden mit einer Grundkonfiguration im KEPServerEXV5 erstellt. Diese können anschliessend im Programm oder im project.xml Dokument an der entsprechenden Stelle manuell abgeändert werden, falls eine spezielle Konfiguration von Nöten sein sollte.

Unity V7

Unity V7 wird verwendet um die SPS von Schneider zu programmieren. Das Format des Dokuments welches verwendet werden muss um `<item>` Elemente in Unity V7 zu importieren ist ein XML Format mit einer .xsy Endung. Diese scheint proprietär zu sein, beinhaltet jedoch eine einfache XML Syntax. Ein solches Dokument sieht wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VariablesExchangeFile>
<fileHeader company="Schneider Automation" product="Unity Pro XL V7.0 - 120823C"
dateTime="date_and_time#2013-6-27-18:40:9" content="Variablen-Quelldatei" DTDVersion="41"/>
<contentHeader name="Projet" version="0.0.8" dateTime="date_and_time#2013-6-27-10:10:0"/>
  <dataBlock>
    <variables name="Input_2" typeName="EBOOL" topologicalAddress="%I1.5.2">
      <attribute name="TimeStampCapability" value="0"/>
      <attribute name="TimeStampSource" value="3"/>
    </variables>
    <variables name="Input_1" typeName="EBOOL" topologicalAddress="%I1.5.1">
      <attribute name="TimeStampCapability" value="0"/>
      <attribute name="TimeStampSource" value="3"/>
    </variables>
    <variables name="Output_1" typeName="EBOOL" topologicalAddress="%Q1.4.1">
      <attribute name="TimeStampCapability" value="0"/>
      <attribute name="TimeStampSource" value="3"/>
    </variables>
    <variables name="StaticWord_1" typeName="WORD" topologicalAddress="%MW120">
      <variableInit value="16#0096"/>
    </variables>
  </dataBlock>
</VariablesExchangeFile>
```

Da eine XSLT das Erstellen eines XML-Dokuments unterstützt, kann ohne weiteres aus einem bestehenden XML-Dokument ein neues XML-Dokument generiert werden.

XSLT Unity

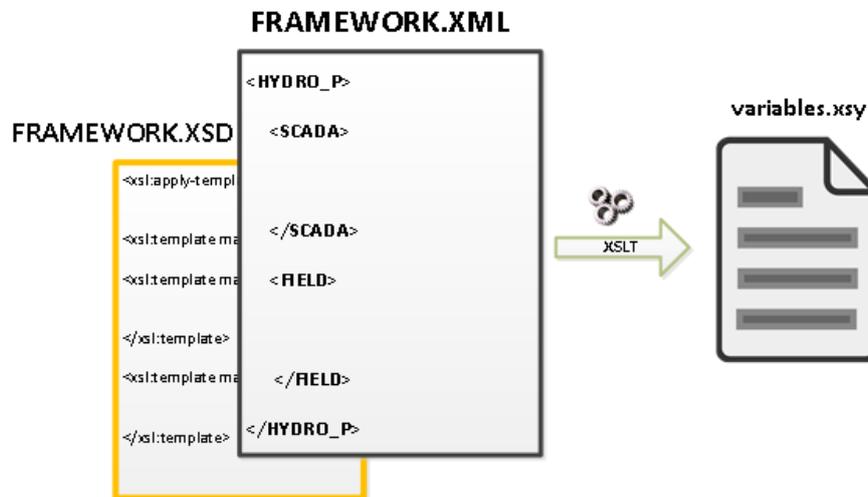


Abbildung 41: XSLTUnity Prinzip

Mittels `<xsl:element name="name of element">value or attribute of current element</xsl:element>` kann ein XML-Element erstellt werden. Das Attribut `name` wird dem XML-Element seinen Namen geben. Innerhalb eines `<xsl:element>` Elements können Attribut-Elemente erstellt werden. Diese sehen wie folgt aus: `<xsl:attribute name="name of attribute">value of attribute</xsl:attribute>`. Das Attribut `name` entspricht dem Namen des Attributs und zwischen dem Attribut kann sein Wert gesetzt werden.

Für jedes `<device>` Element welches in seinem `devicetype` Attribut Modbus stehen hat wird ein dazugehöriges Verzeichnis mit dem zu Importierenden `variables.xsy` XML-Dokument für Unity V7 erstellt. Dieses Verzeichnis trägt den Namen des `name` Attributs des `<device>` Elements.

Die Funktion `<xsl:value-of select="current-dateTime()"/>` gibt die aktuelle Zeit zurück und setzt diese in das Attribut `dateTime` der gewünschten Ausgabe. Im Element `<dataBlock>` wird die Schablonen-Regel für jedes `<item>` Element ausgeführt.

Für jedes `<item>` Element wird dieselbe Struktur erstellt. Diese ändert sich jedoch falls es sich um einen Boolean Datentyp handelt. Dies kann mittels `<xsl:if test="@datatype='Boolean'></xsl:if>` ermittelt werden. Falls diese Bedingung zutrifft wird der sich darin befindende Text, die Bedingung oder das Element ausgeführt. Diese Testabfragen können mehrfach ineinander verschachtelt werden.

Für das Attribut `topologicalAddress` muss eine Bedingung eingeführt werden. Dies resultiert aus dem im Kapitel Beispiel einer erstellten Infrastruktur erwähnten Problem der Adresse eines Items welches einem Modbus Gerät untergeordnet ist. Das verwendete `<xsl:choose>` und `<xsl:when test=""></xsl:when>` kann als eine if else –Bedingung betrachtet werden. Diese kann ebenfalls, wie die `<xsl:if test=""></xsl:if>` Bedingung verschachtelt werden.

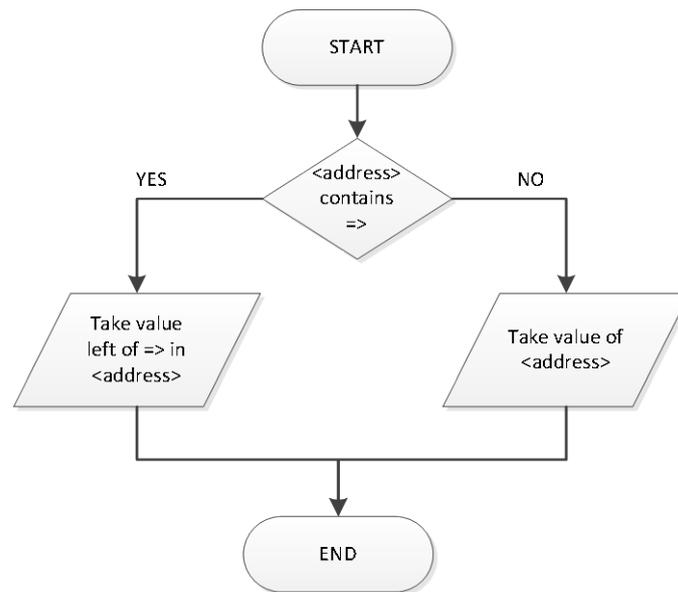


Abbildung 42: <address> Element Trennzeichen Behandlung in Unity V7

Falls ein Trennelement => in dem <address> Element in dem framework.xml Dokument auffindbar ist, wird die linke Seite des <address> Element Wertes verwendet. Falls nicht, wird der ganze Wert, welcher sich in dem <address> Element befindet verwendet.

Für einen Boolean Datentyp müssen ebenfalls Modifikationen vorgenommen werden. Ein Boolean muss in Unity als **EBOOL** Datentyp deklariert werden. Des Weiteren müssen statische Informationen hinzugefügt werden welche *TimeStampCapability* und *TimeStampSource* beinhalten. Das Resultat entspricht dem unter Referenzumgebung & dazugehörige XSLT Unity V7 gezeigten Beispiel XML Code. Daraus kann die Konfiguration der Prozessvariablen für Schneider Geräte aus dem framework.xml Dokument extrahiert werden. Sobald eine Verbindung zu einer SPS besteht kann die erstellte Konfiguration Importiert werden (siehe Kapitel: Test & Validation unter Unity V7). Der ganze Code der XSLT kann im Anhang 9: XSLTUnity.xsl betrachtet werden.

TIA Portal V11

Das Programm TIA Portal V11 von Siemens welches verwendet wird um Siemens S7 SPS zu programmieren, benötigt als Konfigurationsdokument für seine Prozessvariablen, in diesem Kontext PLC Tags genannt, eine Excel Datei (Endung .xlsx). Eine XSLT erlaubt es jedoch nicht direkt eine Excel Datei zu generieren. Deshalb muss ein Zwischenschritt eingelegt werden. Eine XSLT erlaubt es jegliche Art von Text zu erstellen. Deshalb wird ein Textdokument mit der Endung .csv und Semikolon als Trennzeichen erstellt. Dabei muss das Textdokument dasselbe Format wie das von TIA Portal V11 gewünschte Format besitzen (Spalten und Zeilen).

Excel Format:

Name	Path	Data Type	Logical Address	Comment	Hmi Visible	Hmi Accessible
PRSP_VIT_ROT_ARB		Word	%IW272	rack1 slot3	True	True
PRSP_MIP_VIT_95		Boolean	%I0.1	rack1 slot4	True	True
PRSP_MIP_VIT_80		Boolean	%I0.2	rack1 slot5	True	True
PRSP_MIP_VIT_120		Boolean	%I0.3	rack1 slot 6	True	True

Abbildung 43: TIA Portal Zellenformat der Importation

CSV Format:

```
Name;Path;Data Type;Logical Address;Comment;Hmi Visible;Hmi Accessible
PRSP_VIT_ROT_ARB;;Word;%IW272;rack1 slot3;True;True
PRSP_MIP_VIT_95;;Boolean;%I0.1;rack1 slot4;True;True
PRSP_MIP_VIT_80;;Boolean;%I0.2;rack1 slot5;True;True
PRSP_MIP_VIT_120;;Boolean;%I0.3;rack1 slot 6;True;True
```

Ein .CSV Textdokument lässt sich ohne weiteres aus einem XML Dokument generieren.

XSLT TIA Portal V11

Die XSLT um das gewünschte Textdokument zu erhalten sieht folgendermassen aus.

Für jedes `<device>` Element, welches in seinem `devicetype` Attribut S7 stehen hat, wird ein dazugehöriges Verzeichnis mit dem zu importierenden PLC Tags.csv CSV-Dokument für TIA Portal V11 erstellt. Dieses Verzeichnis trägt den Namen des `name` Attributs des `<device>` Elements und wird mit dem bereits gezeigten Codeabschnitt

```
<xsl:variable name="TIAPortal" select="concat(@name, '/', 'PLC Tags', '.csv')" />
<xsl:value-of select="$TIAPortal" /> <!-- Creating -->
  <xsl:result-document href="{ $TIAPortal}" >
    .....
  </xsl:result-document>
```

erstellt.

Dieses Textdokument besitzt einen statischen Abschnitt. Dieser sieht wie folgt aus und wird mit dem Element `<xsl:text></xsl:text>` gekennzeichnet:

Name;Path;Data Type;Logical Address;Comment;Hmi Visible;Hmi Accessible

Da ein `<xsl:text>` Element keine Absätze wahrnimmt müssen diese mittels `` gekennzeichnet werden. Es gibt weitere solcher Sonderzeichen die nicht wahrgenommen werden und deshalb mit einer speziellen Syntax eingefügt werden müssen. Wie z.B

<	<
>	>
&	&
'	'
“	"
New line	

Abbildung 44: XSLT Sonderzeichen

Mit der Schablonen-Regel `<xsl:apply-templates select="//fr:item"/>` wird für jedes gefundene `<item>` Element innerhalb des `<device>` Elements folgender Code ausgeführt:

```
<xsl:value-of select="@name"/>
<xsl:text>;</xsl:text>
<xsl:value-of select="@datatype"/>
<xsl:text>;%</xsl:text>
<xsl:value-of select="fr:address"/>
<xsl:text>;</xsl:text>
<xsl:value-of select="fr:description"/>
<xsl:text>;True;True&#13;</xsl:text>
```

Mittels `<xsl:value-of select="@name"/>` erhält man den Wert des gewünschten Attributs oder Elements. Mit dem Präfix `@` werden Attribute bezeichnet.

Das Ergebnis entspricht dem unter Referenzumgebung & dazugehörige XSLT TIA Portal V11 Format CSV Beispiel.

Durch das Öffnen der .csv Datei in Excel kann die Datei manuell durch Speichern unter... (muss denselben Namen besitzen) in eine .xlsx Datei umgewandelt werden. Sobald mittels TIA Portal V11 Programm eine Verbindung zur SPS erstellt wurde und diese einer Grundkonfiguration (siehe Kapitel: Test & Validation unter TIA Portal V11) unterzogen wurde, kann die erstellte Excel Datei importiert werden.

Der ganze Code der XSLT ist im Anhang 10: XSLTTIAPortal.xml zu finden.

Ergänzende XSLT

Es wurden weitere XSLT erstellt die dem Systemintegrierer unter Cimplicity helfen sollen ein Projekt korrekt zu importieren. Bevor das import_points.csv Dokument importiert werden kann, müssen gewisse Vorkehrungen in Cimplicity getroffen werden:

- Erstellen eines Projekts mit OPC-Client Protokoll
- Erstellen eines Ports mit der Anbindung zum OPC Server
- Erstellen des Geräts „OPC Server“
- Erstellen der OPC Gruppen im master_opc_x.ini Dokument
- Erstellen der Alarm Klassen welche im framework.xml unter den `<point>` Elementen erstellt wurden
- Erstellen der Ressourcen welche den Namen der `<group>` Elemente entsprechen

Für die Alarmklassen, Ressourcen und die OPC Gruppen wurde jeweils eine XSLT erstellt. Diese generieren HTML Dokumente welche besagte Informationen beinhalten. Diese HTML Dokumente lassen sich nach der XSLT im Ordner Cimplicity/httpdocs finden.

XSLT OPC Groups

Um die Eigenschaften einer OPC Gruppe innerhalb von Cimplicity zu definieren müssen diese in einem Konfigurationsdokument eingetragen werden. Es handelt sich dabei um das master_opc_x.ini Dokument. Das `x` steht für den erstellten Port in Cimplicity. Dieses Dokument kann im Cimplicity Projektverzeichnis unter dem data-Ordner gefunden werden. Dort müssen die Von der XSLT generierten OPC Gruppen eingetragen werden. Die XSLT erstellt ein HTML Dokument welches den einzufügenden Text beinhaltet. Für das framework.xml Dokument im Anhang würde das Ergebnis wie folgt aussehen.



Abbildung 45: Resultat der XSLTOPCGroup.xsl

Der Code der XSLT kann im Anhang 11: XSLTOPCGrups.xsl gefunden werden.

XSLT Alarmclasses

Bevor die Points in Cimplicity importiert werden können müssen deren Alarmklassen erstellt werden. Um diesen Prozess zu vereinfachen wurde eine XSLT erstellt welche in einem HTML Dokument die zu erstellenden Alarmklassen anzeigt.

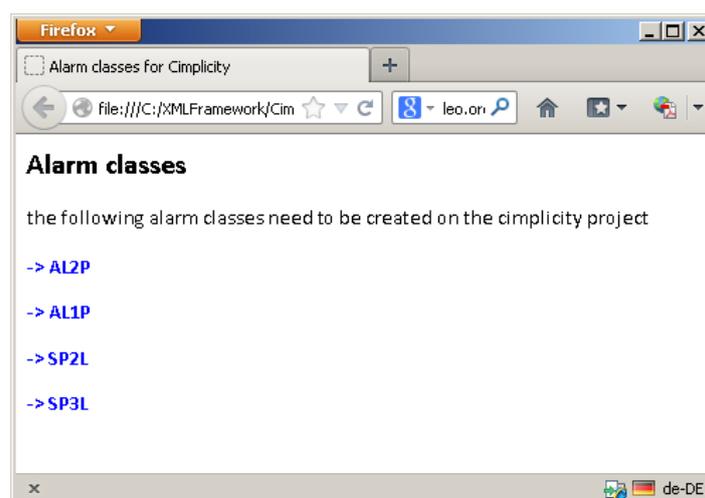


Abbildung 46: Resultat der XSLTAlarmclasses

Der Code der XSLT kann im Anhang 12: XSLTAlarmclasses.xsl gefunden werden.

XSLT Resources

Bevor die Points in Cimplicity importiert werden können müssen deren Ressourcen erstellt werden. Um diesen Prozess zu vereinfachen wurde eine XSLT erstellt welche in einem HTML Dokument die zu erstellenden Ressourcen anzeigt.

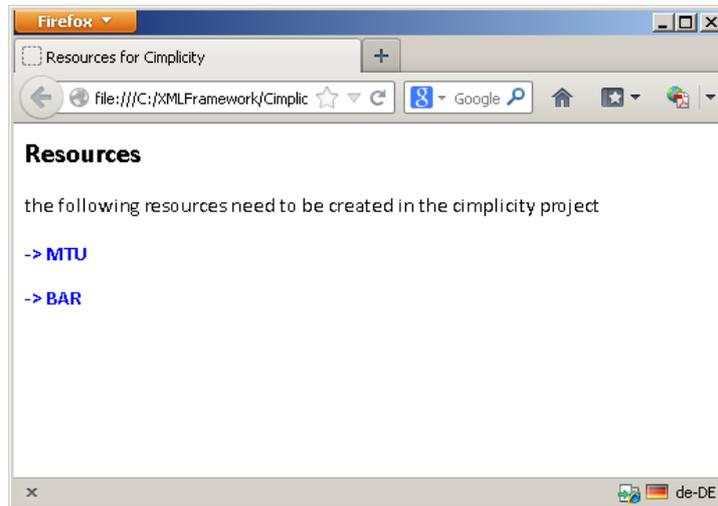


Abbildung 47: Resultat der XSLTResources

Der Code der XSLT kann im Anhang 13: XSLTResources.xsl gefunden werden

Test & Validation

Das erstellte framework.xml Dokument wird allen XSL Transformationen unterzogen.

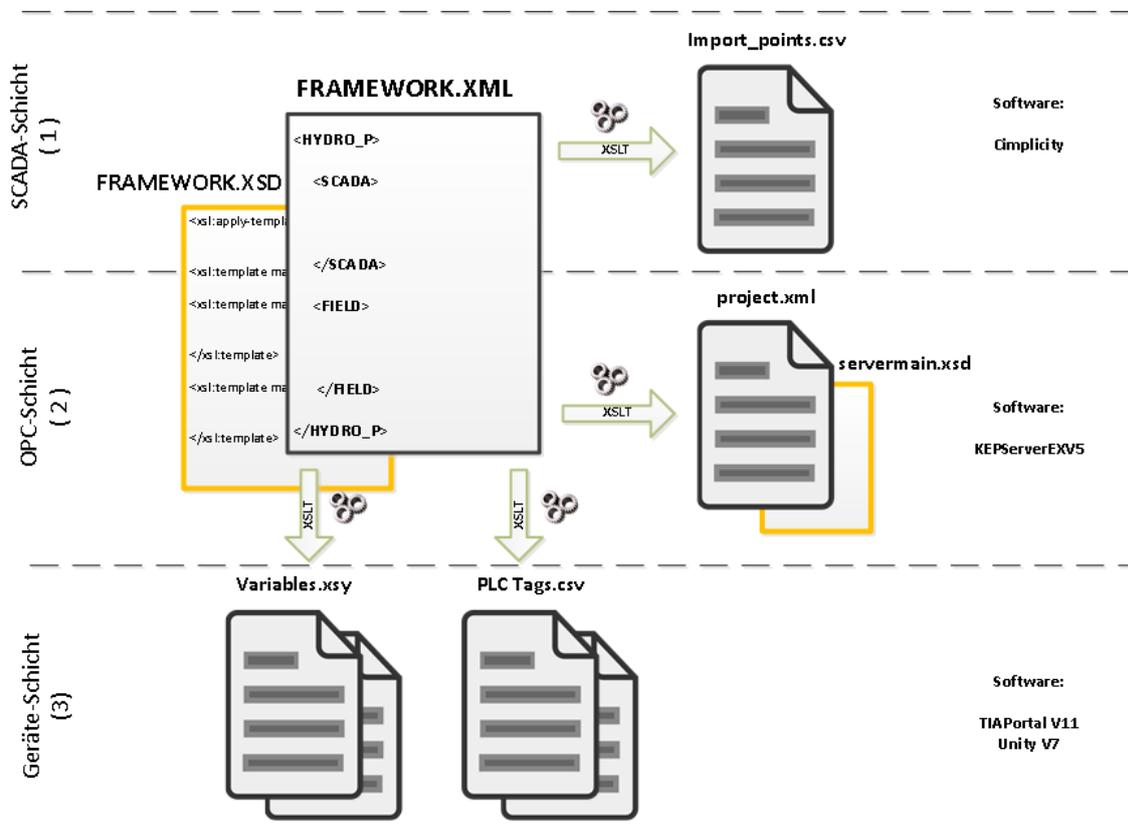


Abbildung 48: XSLT Transformationen für die entsprechenden Schichten

Für jedes Gerät wird ein Verzeichnis mit dessen Namen erstellt. Handelt es sich um ein Modbus Gerät, befindet sich in seinem Verzeichnis ein **variable.xsy** XML-Dokument das alle dazugehörigen Prozessvariablen beinhaltet. Handelt es sich um ein Siemens Gerät, befindet sich darin ein **PLC Tags.csv** Dokument welches alle dazugehörigen Prozessvariablen enthält. Unter dem Verzeichnis Kepware wird ein **project.xml** Dokument erstellt, welches zur Konfiguration des KEPServerEXV5 Programms verwendet wird. Innerhalb des **project.xml** Dokuments befinden sich dieselben Namen der Prozessvariablen wie unter den Konfigurationsdokumenten der Geräte-Schicht. Alle im **framework.xml** erstellten **<agent>**, **<device>**, **<collection>** und **<item>** Elemente entsprechen den in der Abbildung 34: KEPServerEX V5 Design Prinzip. Innerhalb des Verzeichnisses Cimplicity/Points befindet sich das generierte **import_points.csv** Dokument. Dieses beinhaltet alle Prozessvariablen welche im **<scada>** Abschnitt des **framework.xml** Dokuments mit dem **<field>** Abschnitt referenziert wurden. Die 3 zusätzlichen XSL Transformationen um Cimplicity für die Importation der Points vorzubereiten erstellen die korrekten HTML Dokumente im Cimplicity/httpdocs Verzeichnis.

TIA Portal V11

Nach der Verbindung zur SPS mittels TIA Portal V11 und einer minimalen Konfiguration der angeschlossenen Karten lässt sich das entsprechende PLC Tags.xlsx Dokument importieren.

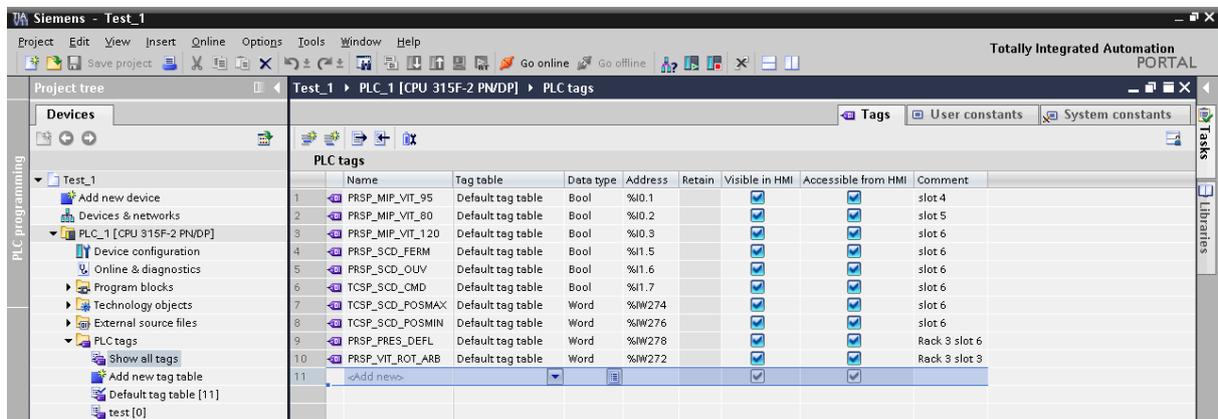


Abbildung 49: Resultat der Importation unter TIA Portal V11

Man erkennt, dass alle Prozessvariablen des entsprechenden Geräts innerhalb des framework.xml Dokuments wieder auffindbar sind. Die Importation für Siemensgeräte funktioniert somit korrekt und entspricht den Erwartungen.

Unity V7

Nach der Verbindung zur SPS mittels Unity V7 und einer minimalen Konfiguration der angeschlossenen Karten lässt sich das entsprechende variables.xsy Dokument importieren.

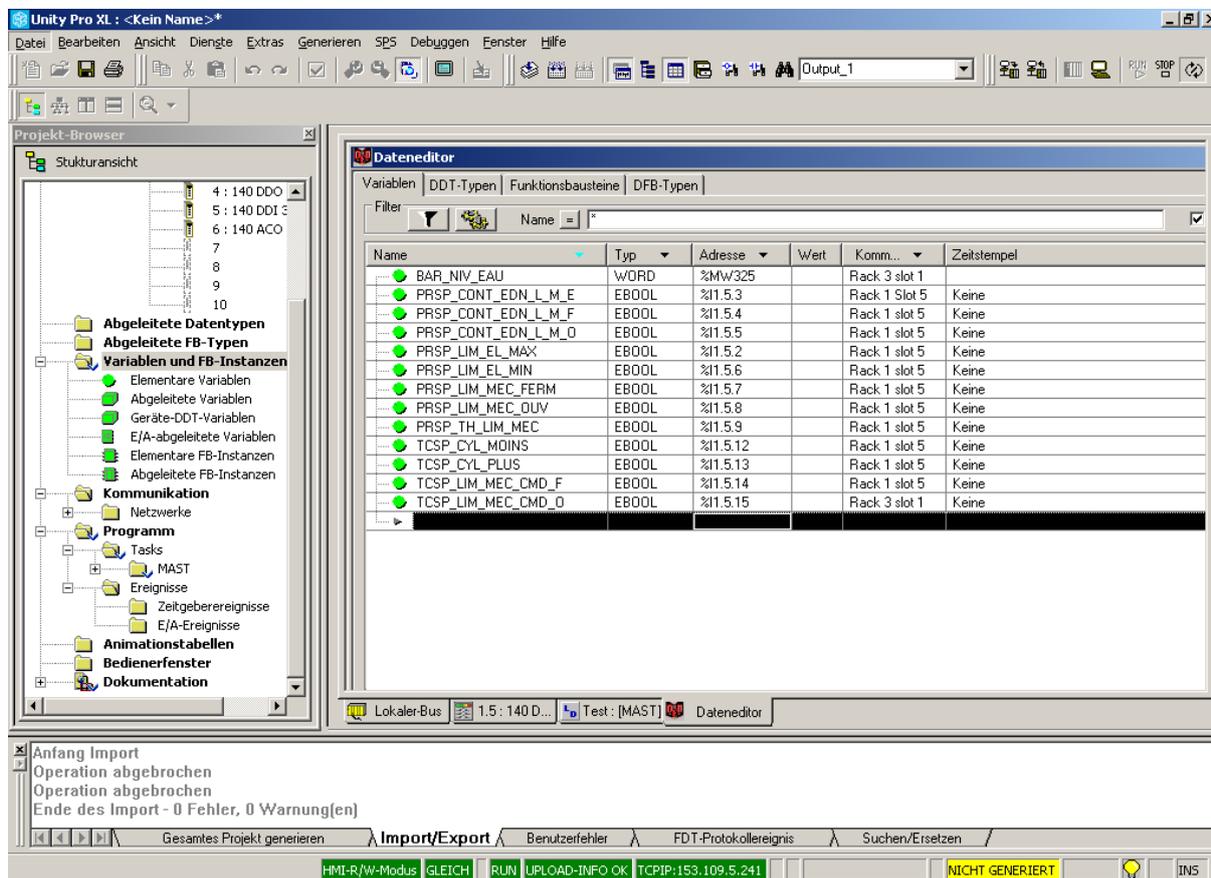


Abbildung 50: Resultat der Importation unter Unity V7

Man erkennt, dass alle Prozessvariablen des entsprechenden Geräts innerhalb des framework.xml Dokuments wieder auffindbar sind. Um die Richtigkeit der Modbusadressen und deren Ein/Ausgänge zu gewährleisten muss der interne Speicher der digitalen Ausgangskarte kontrolliert werden.

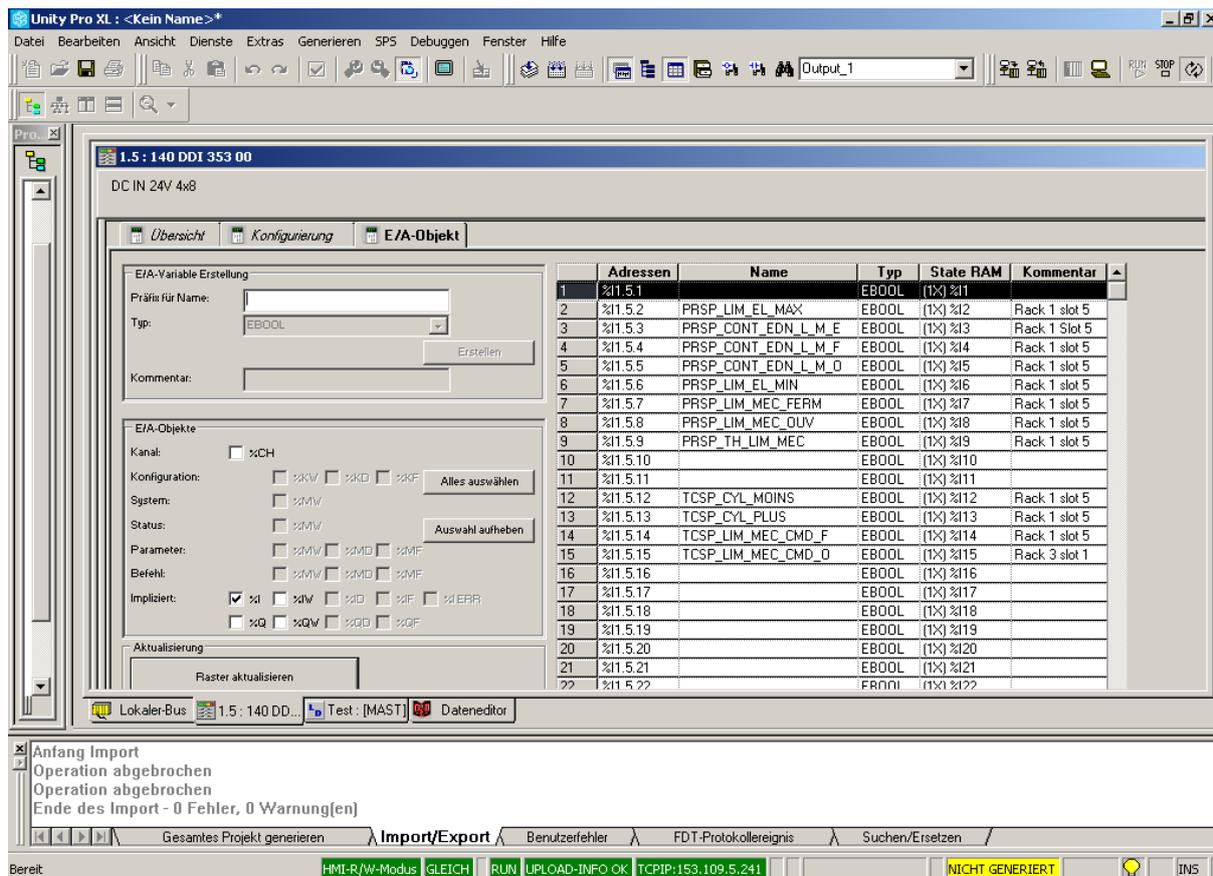


Abbildung 51: Interne Speicheradressierung der digitalen Eingangskarte

Die definierten Ein/Ausgänge eines Geräts welches mit Unity V7 programmiert wird, muss innerhalb des framework.xml Dokuments mit den entsprechenden internen Speicherpositionen übereinstimmen. Das Ergebnis entspricht den Erwartungen man sieht, dass für jede Adresse %x.x.x die korrekte interne Speicheradresse zu finden ist welche einer Modbusadresse entspricht. Das Ergebnis entspricht den Erwartungen und funktioniert korrekt.

Kepware

Sobald das project.xml Dokument aus dem framework.xml Dokument generiert wurde kann die kepwareconfig.bat Datei ausgeführt werden. Diese lädt das erstellte Projekt automatisch in den KEPServerEXV5 OPC Server.

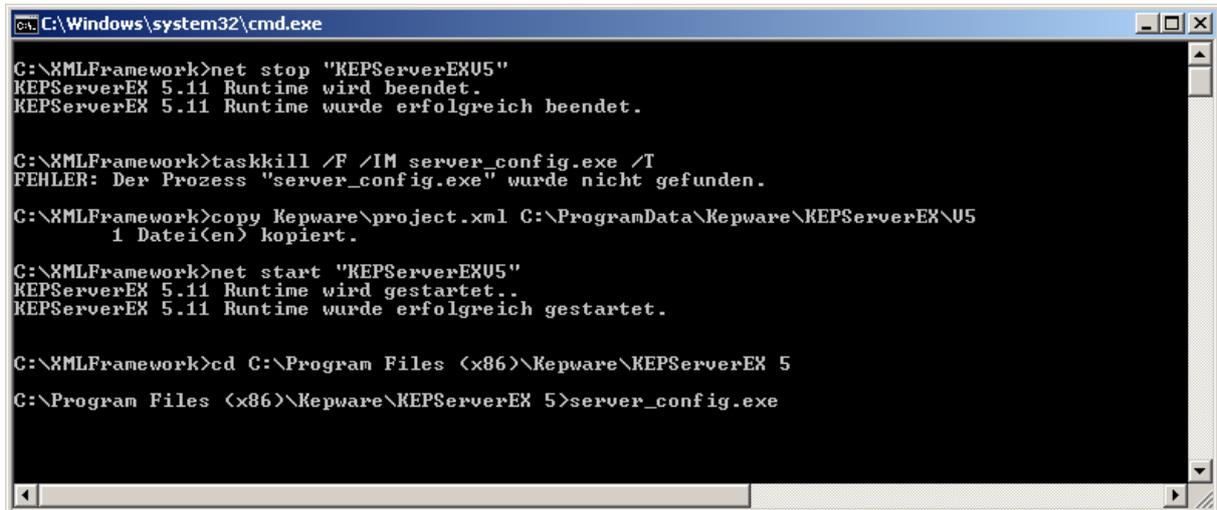


Abbildung 52: Ausführung des kepwareconfig.bat Dokuments

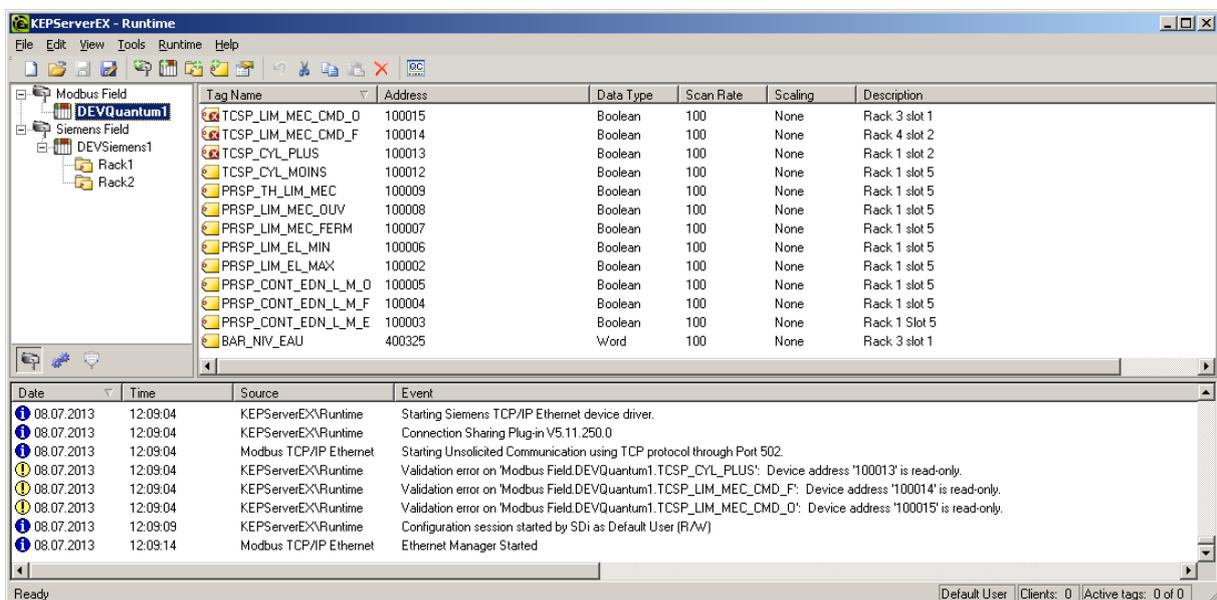


Abbildung 53: Kepware Modbusgerät DEVQuantum1 mit seinen Prozessvariablen

Man erkennt in dem Logfenster unten dass 3 Adressen falsche beziehungsweise unzulässige Zugriffsrechte innerhalb des framework.xml Dokuments hatten. Diese müssen korrigiert werden damit man mit Ihnen arbeiten kann, da man laut Kepware auf Modbuseingänge nur lesen kann und auf diesen nicht schreiben darf.

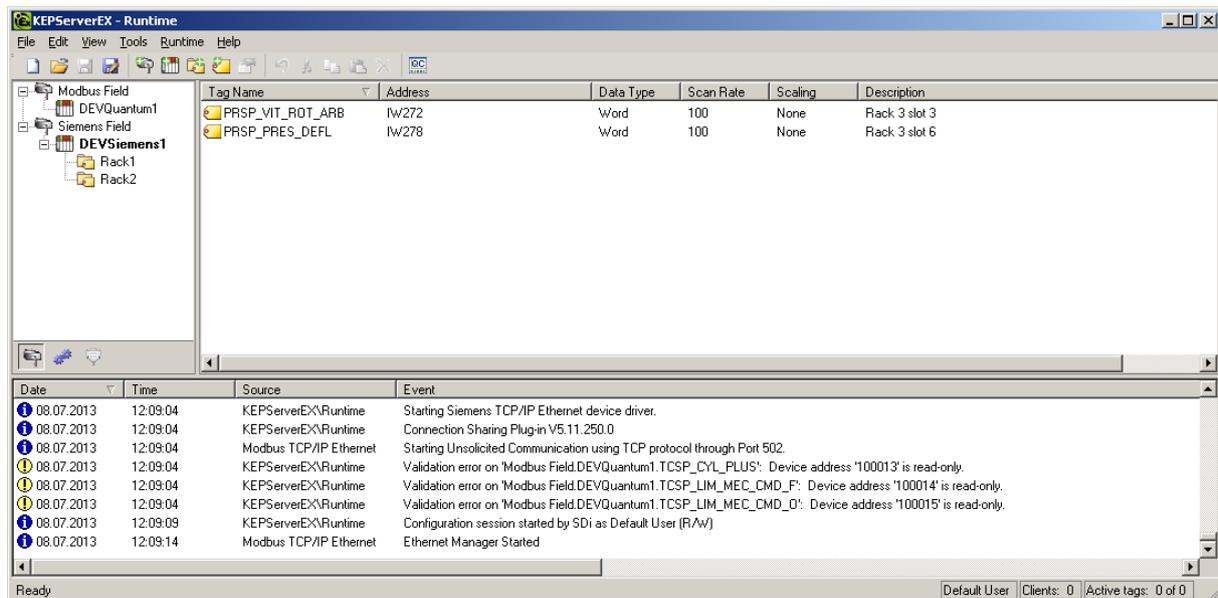


Abbildung 54: Kepware Siemensgerät DEVSiemens1 mit seinen Prozessvariablen

Man erkennt, dass das Siemens Gerät zwei Untergruppen besitzt (Rack1 und Rack2). Diese entsprechen den `<collection>` Elementen im entsprechenden Gerät. Darin befinden sich die untergeordneten Prozessvariablen. Die Konfiguration des OPC Servers funktioniert und arbeitet korrekt. Das Ergebnis entspricht den Erwartungen.

Cimplicity

Nachdem ein neues Projekt in Cimplicity erstellt wurde (OPC Client Protokoll) muss ein Port und anschliessend ein entsprechendes Gerät hinzugefügt werden. Das Gerät entspricht dabei dem OPC Server (Kepware). Dieser besitzt folgende Adresse:

`KEPWARE.KEPServerEX.V5`

Diese Adresse variiert anhand des Standorts des Servers. Da zurzeit keine DCOM Konfiguration vorgenommen wurde bleibt der OPC Server mit besagter Adresse nur lokal zugänglich.



Abbildung 55: Name des Device unter Cimplicity

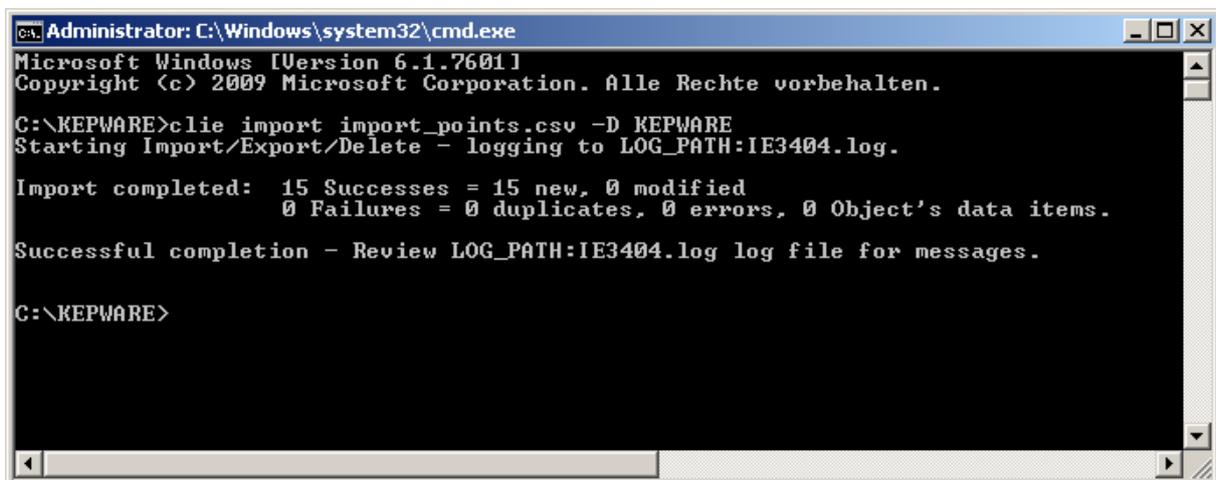
Der Name des Device muss später verwendet werden um den zu importierenden Points ein Gerät zuzuweisen.

Anhand der sich in Cimplicity/httpdocs befindenden HTML Dokumenten können die benötigten Alarmklassen, Ressourcen und OPC Gruppen in Cimplicity erstellt werden.

Nachdem das import_points.csv Dokument im Projekt Verzeichnis kopiert wurde, kann durch einen Konsolenbefehl die Liste Importiert werden.

```
clie import import_points.csv -D KEPWARE
```

Der Parameter -D wird verwendet um dem import.csv Dokument in der Kolonne Device den übergebenen Wert (KEPWARE entspricht dem in der Abbildung 55: Name des Device unter Cimplicity gesetzten Wert) zuzuordnen, falls dieser keinen anderen Wert beinhaltet. Das generierte import_points.csv Dokument besitzt standartmässig keinen Eintrag innerhalb der Device Kolonne, deshalb ist dieser Wert ausschlaggebend und muss dem Befehl hinzugefügt werden ansonsten können die Points nicht importiert werden.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\KEPWARE>clie import import_points.csv -D KEPWARE
Starting Import/Export/Delete - logging to LOG_PATH:IE3404.log.

Import completed: 15 Successes = 15 new, 0 modified
                  0 Failures = 0 duplicates, 0 errors, 0 Object's data items.

Successful completion - Review LOG_PATH:IE3404.log log file for messages.

C:\KEPWARE>
```

Abbildung 56: Cimplicity Importation der Points

Man sieht, dass keine Fehler bei der Importation vorgefallen sind.

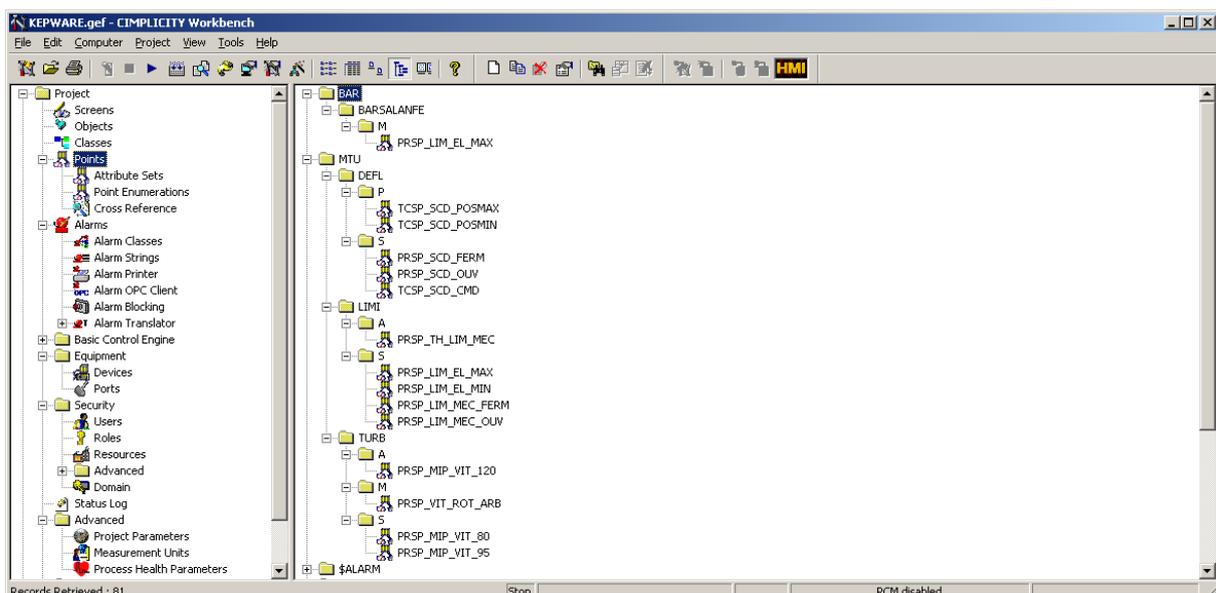


Abbildung 57: Importierte Points in Cimplicity

Alle Points wurden korrekt und in Ihrer vorgegebenen Struktur in Cimplicity Importiert. Alle sich unter MTU befindenden Points unterliegen der MTU OPC Gruppe und der MTU Resource.

Man erkennt dieselben Prozessvariablen wie in den anderen Programmen wieder, die Kohärenz ist somit gewährleistet. Das Ergebnis der Importation entspricht den Erwartungen.

Kepware mit den angeschlossenen Feldgeräten

Öffnet man im KEPServerEXV7 den OPC Quick Client kann man die Kommunikation zwischen den Feldgeräten und dem OPC Server testen. Die Prozessvariablen der angeschlossenen Feldgeräte und deren Werte können auf beiden Seiten kontrolliert werden.

	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		"TCSP_SCD_P...	%IW274:P	DEC_signed	9144		<input type="checkbox"/>	
2		<Add new>					<input type="checkbox"/>	

Abbildung 58: TIAPortalV11 Wert der Prozessvariable TCSP_SCD_POSMAX

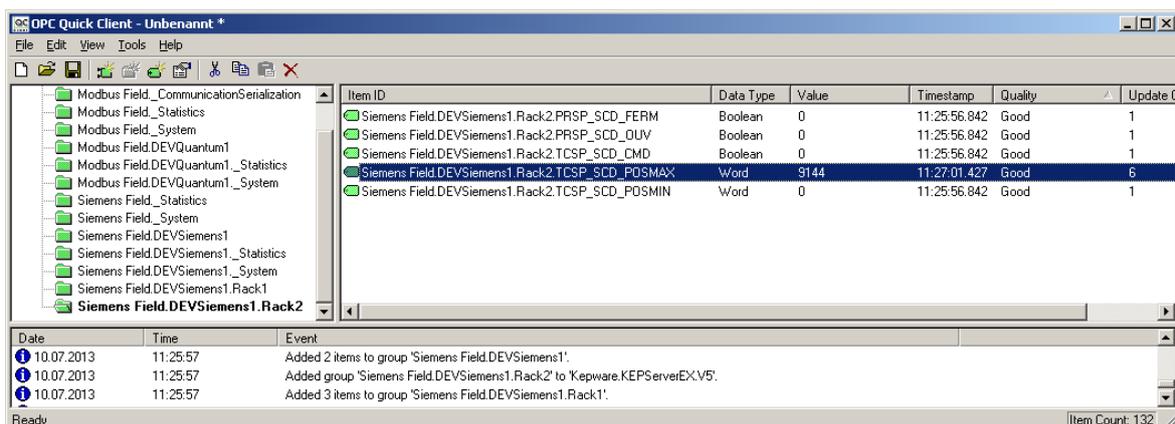


Abbildung 59: KEPServerEXV5 Wert der Prozessvariable TCSP_SCD_POSMAX

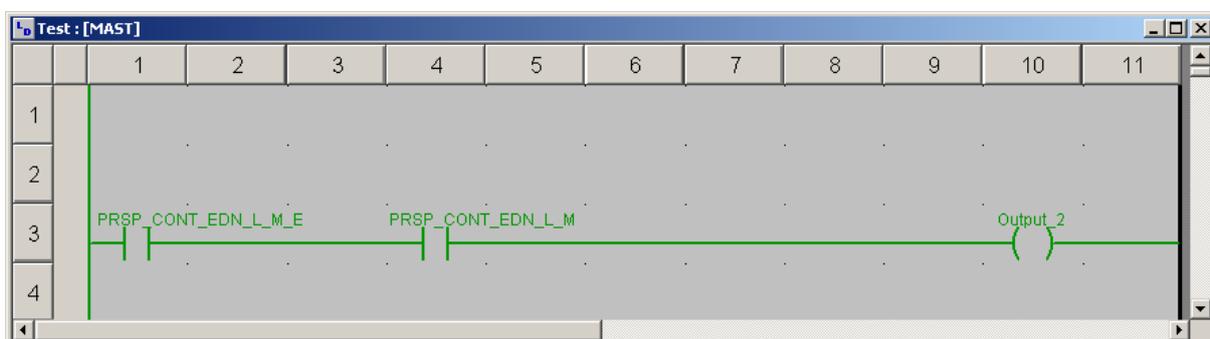


Abbildung 60: Unity V7 Wert der Prozessvariablen PRSP_CONT_EDN_L_M_E und PRSP_CONT_EDN_L_M_F

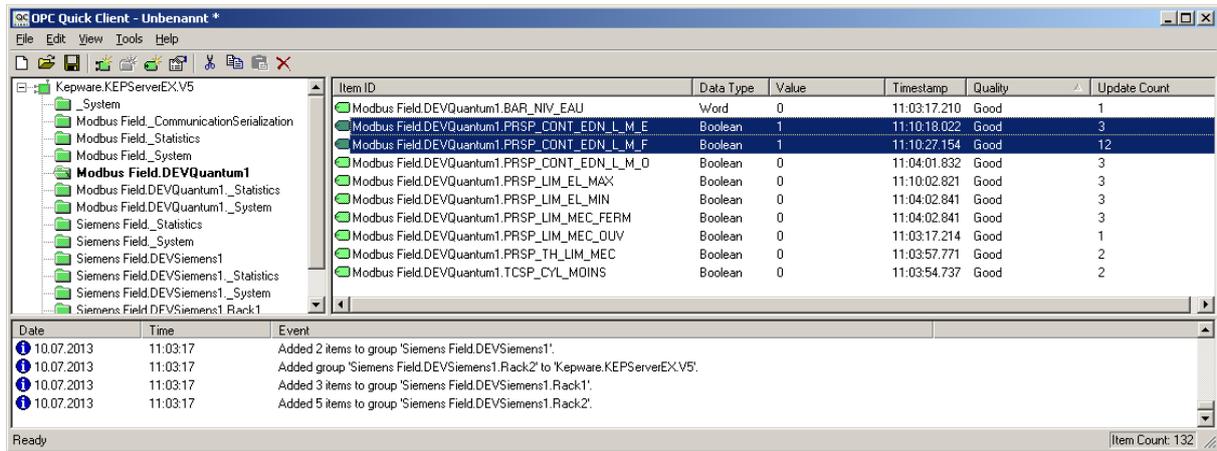


Abbildung 61: KEPServerEXV5 Werte der Prozessvariablen PRSP_CONT_EDN_L_M_E und PRSP_CONT_EDN_L_M_F

Die Resultate Entsprechend den Erwartungen die physikalischen Eingänge der SPS werden im OPC Server gespiegelt und können von dem sich in KEPServerEXV5 befindenden OPC Client gelesen werden.

Kepware mit dem Cimplicity OPC Client

Durch den Point Control Panel in Cimplicity kann man den aktuellen Wert der Einträge konsultieren. Es handelt sich dabei um einen Einfachen OPC Client der die Werte auf dem OPC Server liest.

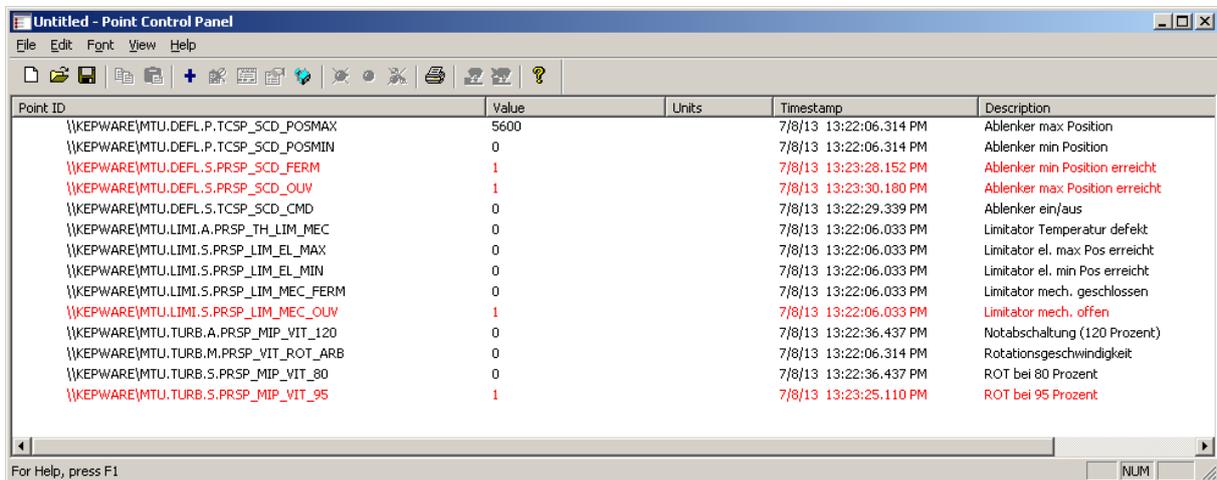


Abbildung 62: OPC Client in Cimplicity

Die rot markierten Einträge entsprechen Einem Alarm oder einer Signalisation.

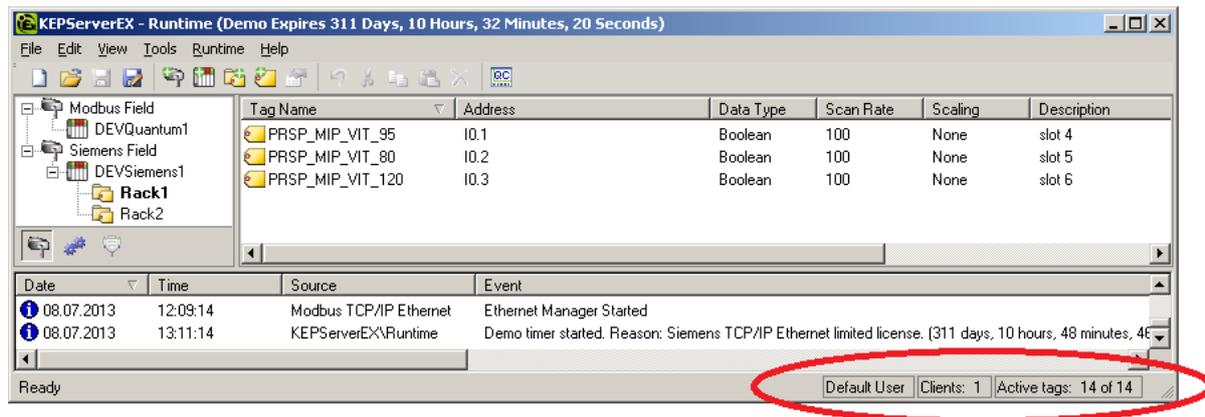


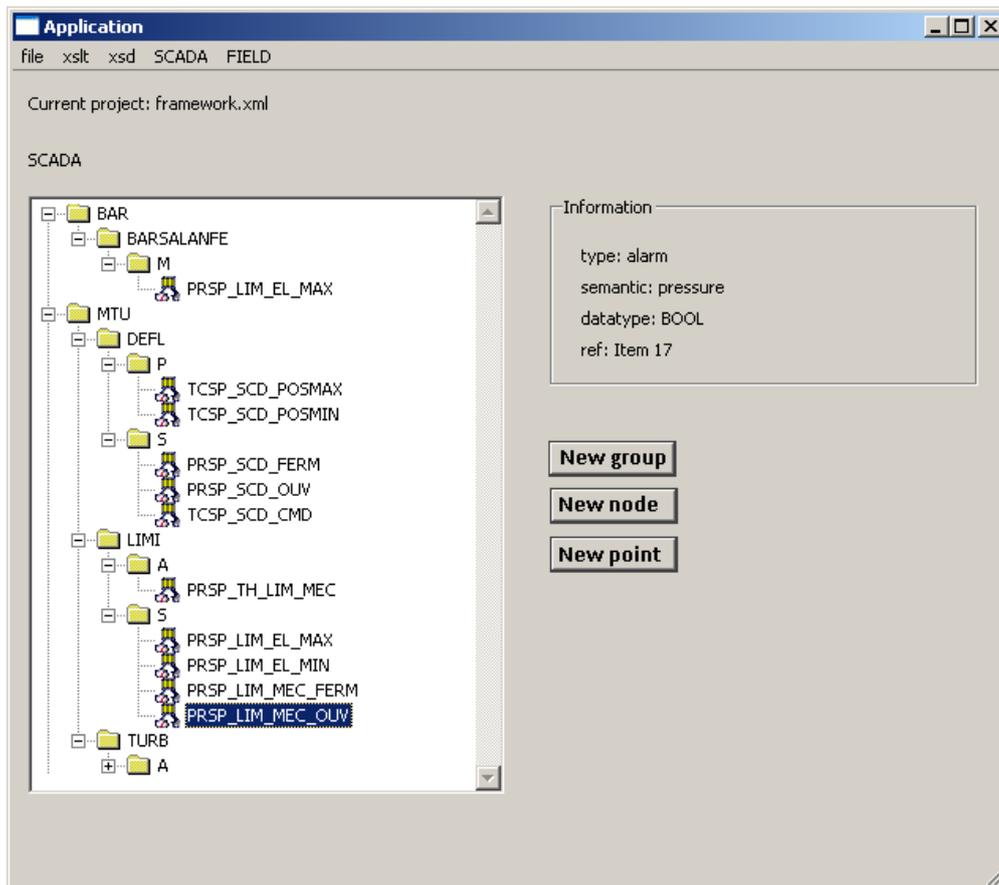
Abbildung 63: Verbundene OPC Clients auf dem Kepware OPC Server

Auf dem Kepware OPC Server erkennt man, dass ein OPC Client auf dem Server verbunden ist. Die Resultate Entsprechend den Erwartungen die physikalischen Eingänge der SPS werden im OPC Server gespiegelt und können von dem sich in Cimplicity befindenden OPC Client gelesen werden.

Benutzeroberfläche

Zurzeit wird das framework.xml Dokument mit dem oXygen XML Editor bearbeitet. Es wäre möglich in einem zukünftigen Projekt eine grafische Oberfläche basierend auf dem XML beziehungsweise XSD Dokument zu entwickeln welches den Planungsprozess eines Wasserkraftwerks weiter vereinfachen würde. Die Umsetzung der grafischen Oberfläche könnte in verschiedenen Programmiersprachen umgesetzt werden.

Eine Interessante Lösung wäre eine webbasierte Benutzeroberfläche zu erstellen. Diese könnte auf dem bestehenden XML oder gar XSD Dokument aufbauen und die Handhabung des Mapping zwischen dem `<field>` und dem `<scada>` Abschnitt deutlich vereinfachen. Des Weiteren könnte eine Datenbank generiert werden welche ebenfalls alle Informationen des framework.xml Dokuments beinhaltet. Mittels Datenbank und Webapplikation könnte eine Unabhängigere Arbeitsweise garantiert werden. Zum Beispiel könnte eine Person den SCADA Abschnitt modellieren und eine weitere Person zur selben Zeit den FIELD Abschnitt. Dabei müssten sie nicht mehr ein XML Dokument hin und her senden um es zu modifizieren. Ein mögliches Design könnte wie folgt aussehen:



Information

type: alarm
 semantic: pressure
 datatype: BOOL
 ref: Item 17

- New group
- New node
- New point

Abbildung 64: Mögliches Design für den SCADA Abschnitt

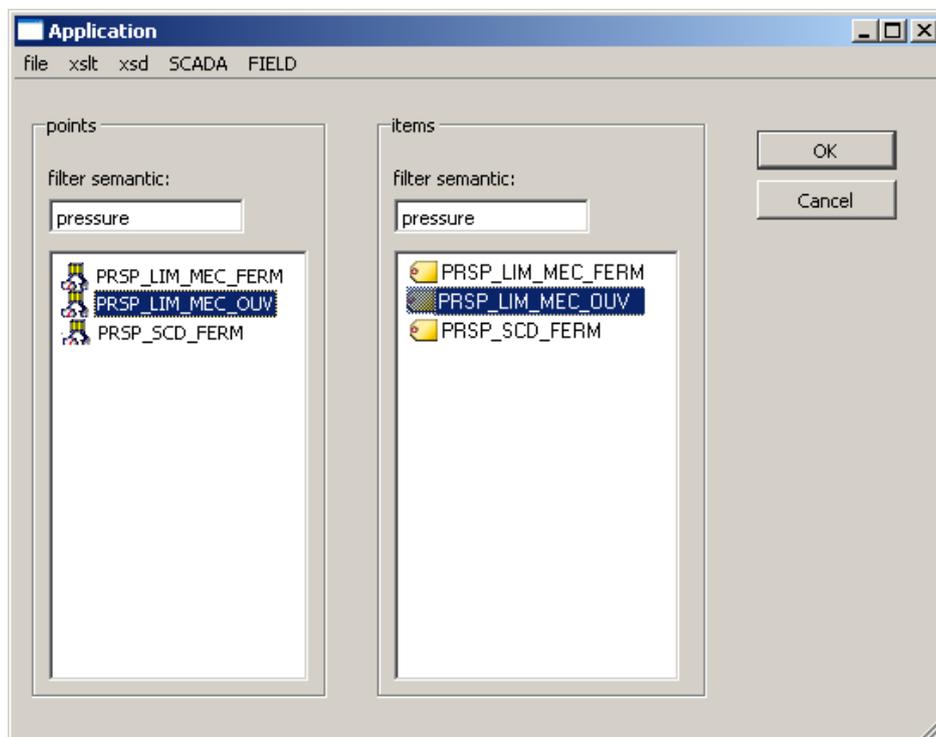


Abbildung 65: Mögliches Design für das Mapping der points mit den items

Schlussfolgerung

Es wurde ein Prozess entwickelt welcher den bestehenden Prozess der Planung, Erneuerung und Implementierung einer Wasserkraftwerkanlage von HYDRO-Exploitation unabhängig seiner Architektur und seines Aufbaus optimiert. Das Ziel ein einheitliches Verfahren zur entwickeln welches eine Kohärenz der Prozessvariablen durch alle Schichten (SPS, OPC und SCADA) besitzt wurde erreicht. Es wurde eine Richtlinie entworfen welche OPC Elemente (items) innerhalb eines OPC Servers und in den Feldgeräten definiert. Die Definition von OPC Gruppen (OPCGroup) in einem OPC Server und im OPC-Client wurde ebenfalls in der Richtlinie entworfen. Die Richtlinie wurde in einem UML Diagramm entworfen und mittels XSD Dokument umgesetzt. Das XSD Dokument erstellt eine Grundstruktur welche beschreibt wie ein darauf aufgebautes XML Dokument auszusehen hat. Ein Testanlage mit:

- eine SPS von Siemens (S7-300)
- eine SPS von Schneider (Quantum)
- einem OPC Server von Kepware (KEPServerEXV5)
- einem OPC Client innerhalb von einem SCADA Programm (Cimplicity)

wurde in Betrieb genommen und getestet. Aus einem erstellten XML Dokument basierend auf der XSD Richtlinie wurden alle Konfigurationen der Programme der Testanlage extrahiert. Der dabei verwendete Prozess heisst XSL Transformation. Nach der Konfiguration der angeschlossenen Geräte mit den dazugehörigen extrahierten Konfigurationen wurden Funktionstests durchgeführt. Diese entsprachen den Erwartungen.

Einzig das Zeitstempel Teilziel und das IEC 61850 Geräte und Implementationsziel wurde nicht erreicht. Diese beiden Ziele wurden im Verlauf des Projekts aus den zu erfüllenden Zielen und Teilzielen entfernt. Das Zeitstempel Teilziel beinhaltet das von HYDRO-Exploitation verwendete Prinzip zur Handhabung der Zeitstempel unter Quantum Geräte.

Die Zukunft des Projekts sieht vor evtl. eine graphische Benutzeroberfläche zu erstellen welche auf dem XML und XSD Dokument aufbaut. Des Weiteren müssen noch andere Protokolle der bestehenden Umgebung implementiert und getestet werden. Eine weit entfernte Vision wäre, dass ein OPC Client entwickelt werden würde der in einem eigens entwickelten SCADA System integriert wäre. Somit hätte HYDRO-Exploitation ein eigenes auf OPC basierendes SCADA System welches den Anforderungen von HYDRO-Exploitation entsprechen würde.

Literaturverzeichnis

- [1] Unbekannt, «Wikipedia,» [Online]. Available: http://de.wikipedia.org/wiki/Supervisory_Control_and_Data_Acquisition. [Zugriff am 9 July 2013].
- [2] Unbekannt, «Wikipedia,» [Online]. Available: http://de.wikipedia.org/wiki/Extensible_Markup_Language. [Zugriff am 9 Juli 2013].
- [3] Unbekannt, «Wikipedia,» [Online]. Available: <http://de.wikipedia.org/wiki/XSD>. [Zugriff am 9 Juli 2013].
- [4] Unbekannt, «Wikipedia,» [Online]. Available: <http://de.wikipedia.org/wiki/XSLT>. [Zugriff am 9 Juli 2013].
- [5] O. T. I. (OPCTI), OPC & DCOM Diagnostics Hands-On OPC Training - Level 1, 2010.
- [6] Norm IEC61850

Datum

Sitten, der 12 Juli 2013

Unterschrift

Pascal Mengis

Anhang

Anhang 1	OPCBericht.pdf	nur elektronisch auf der CD zu finden
Anhang 2	LIST import.xlsx	ganzes Dokument nur auf der CD zu finden
Anhang 3	framework.xsd	
Anhang 4	Gr3 Liste superviseur.xls	ganzes Dokument nur auf der CD zu finden
Anhang 5	framework.xml	
Anhang 6	XSLTCimplicity.xsl	
Anhang 7	servermain.xsd	nur elektronisch auf der CD zu finden
Anhang 8	XSLTKepware.xsl	nur elektronisch auf der CD zu finden
Anhang 9	XSLTUnity.xsl	
Anhang 10	XSLTTIAPortal.xsl	
Anhang 11	XSLTOPCGroups.xsl	
Anhang 12	XSLTAlarmclasses.xsl	
Anhang 13	XSLTResources.xsl	

Anhang 1

BerichtOPC.pdf

Studiengang Systemtechnik

Vertiefungsrichtung Infotronik

Semesterprojekt 2013

Pascal Mengis

OPC Server

Dozent Dominique Gabioud

Sitten, 29. April 2013

INHALTSVERZEICHNIS

1	OPC Server.....	1
1.1	<i>Einleitung</i>	<i>1</i>
1.2	<i>Beispiel : OPC Server Architektur.....</i>	<i>1</i>
1.3	<i>Simulationsumgebung.....</i>	<i>2</i>
2	OPC Schnittstellenspezifikation.....	4
2.1	<i>Einleitung</i>	<i>4</i>
2.2	<i>OPC Item Objekt.....</i>	<i>5</i>
2.3	<i>OPC Server Objekt</i>	<i>5</i>
2.4	<i>OPC Client Objekt.....</i>	<i>7</i>
2.5	<i>OPC Gruppen Objekt.....</i>	<i>9</i>
3	Evaluation	14
3.1	<i>Einleitung</i>	<i>14</i>
3.2	<i>Kriterienliste</i>	<i>14</i>
3.3	<i>Ergebnis.....</i>	<i>15</i>
4	Schlussfolgerung	17
5	Bibliography.....	18
6	Anhang	18

EINFÜHRUNG IN DA (DATA ACCES) EINES OPC SERVERS

1 OPC SERVER

1.1 Einleitung

OPC ist eine standardisierte Schnittstelle die den Zugriff auf Prozessdaten in Echtzeit ermöglicht. Sie basiert auf Microsoft (DCOM) und dient zum Lesen und Schreiben von Prozessvariablen innerhalb eines Automationsnetzwerks. OPC ist ein weitverbreiteter und oft verwendeter Standard in der Industrie. Er zeichnet sich vor allem durch seine Organisation der Daten aus. Ein wesentlicher Vorteil von OPC zu bestehenden Systemen besteht darin, dass die Daten strukturiert werden und somit einfacher anderen Schichten (Verwaltung, HMI, SCADA usw.) zur Verfügung stehen. Durch die standardisierte Schnittstelle kann ein Automationsnetzwerk weit aus einfacher organisiert und betrieben werden.

1.2 Beispiel : OPC Server Architektur

Betrachtet man ein Grossunternehmen welches eine Fertigungshalle und ein Büro besitzt kann die Implementierung von OPC wie folgt verstanden werden:

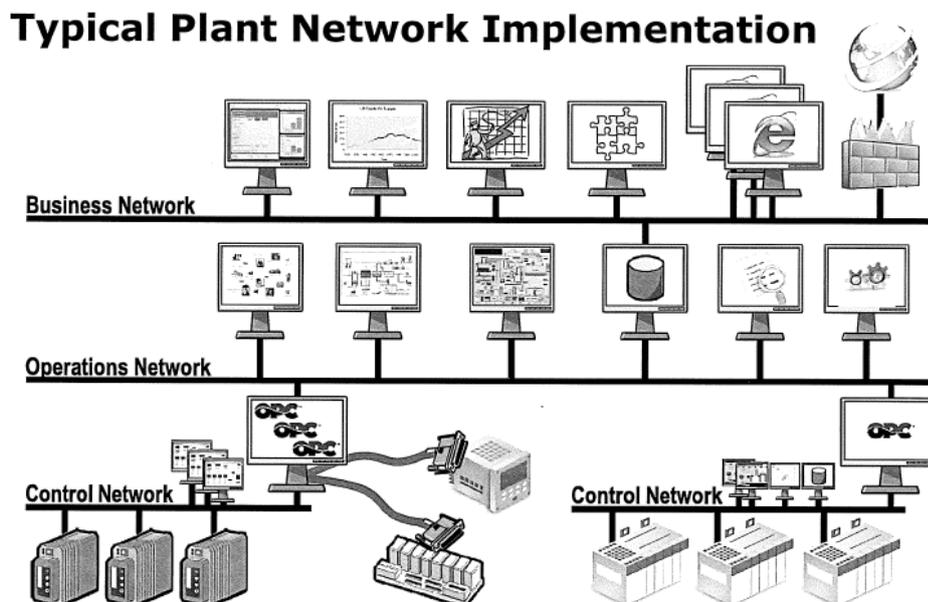


Abbildung 1 Typische OPC Implementation [1]

Business Netzwerk (Rechner im Büro des Grossunternehmens)

Besitzt Applikationen wie z.B. Statistikprogramme welche zur Prozessoptimierung ebenfalls Zugriff auf den Daten des Prozesses haben möchten.

Operation Netzwerk (Rechner in der Fertigungshalle des Grossunternehmens)

Besitz Applikationen wie z.B. HMI's (Human Machine Interface) oder SCADA's welche direkten Zugriff auf Daten des Automationsnetzwerks haben möchten.

Kontroll Netzwerk (Rechner & Geräte in der Fertigungshalle des Grossunternehmens)

Dient der Regelung des Prozesses auf den Speicherprogrammierbaren Geräten welche unter verschiedenen Feldbusse wie z.B. ProfiBus, Modbus, TCP/IP usw. verbunden sind.

1.3 Simulationsumgebung

In diesem Semesterprojekt werden nicht alle Aspekte von OPC behandeln sondern nur der vom OPC DA (Data Access).

Um die Funktionsweise von OPC DA besser zu verstehen wurde eine Simulationsumgebung auf einem Rechner installiert. Die Simulationsumgebung beinhaltet zwei OPC Server von unterschiedlichen Herstellern und deren dazugehörigen Client Anwendungen.

- Softing
- Matrikon

Simulationsumgebung

- Betriebssystem
 - Windows7
- Server OPC
 - MatrikonOPC Server for Simulation
 - Softing S7/S5 OPC Server
- Clients
 - MatrikonOPC Explorer
 - Softing OPC Toolbox Demo Client
- Siemens S7 Simulator
 - DELTALOGIC : ACCONtrol S7

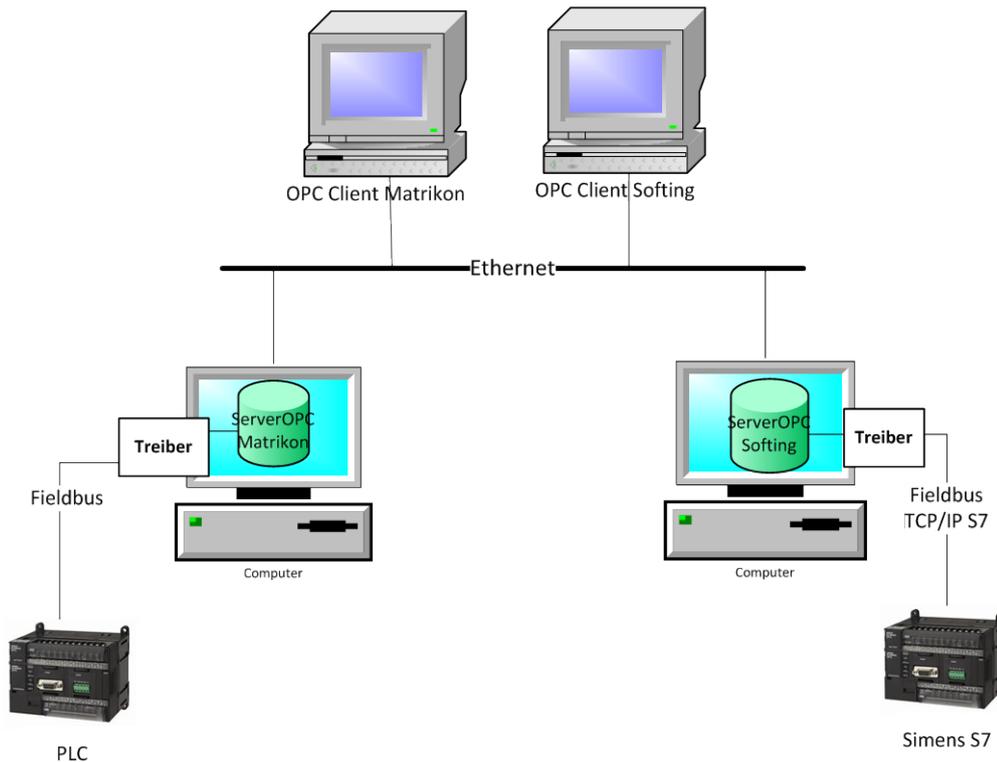


Abbildung 2 Simulationsumgebung einer OPC Server Architektur

Der Testserver von Matrikon simuliert bereits intern ein Feldbus mit angeschlossenen Geräten (PLC's: Programming Logic Control). Der Softing OPC Server jedoch überlässt dem Anwender die Freiheit diverse Geräte (in unserem Fall: Simulatoren von Simens S7 Geräten auf TCP/IP Basis) anzuschliessen. Die besagte Simulationsumgebung befindet sich innerhalb eines Computers und dient rein zu testzwecken um ein besseres Verständnis von OPC zu erhalten. Alle vorgenommenen Tests wie zum Beispiel:

- das Erstellen einer Gruppe
- das Durchsuchen des Servers nach Items
- das Schreiben auf einem oder mehreren Items
- das Lesen von einem oder mehreren Items

wurden verwendet um das Verhalten der OPC Schnittstelle besser zu verstehen.

2 OPC SCHNITTSTELLENSPEZIFIKATION

2.1 Einleitung

Um ein fundamentaleres Verständnis der Vorgänge innerhalb eines OPC Servers zu erhalten, können anhand der offiziellen OPC Spezifikation heraus Objektdiagramme und Sequenzdiagramme erstellt werden. Diese Objekte und Sequenzdiagramme lassen eine objektorientierte Ansicht eines OPC Servers zu. Die Softwarearchitektur lässt sich vereinfacht wie folgt darstellen:

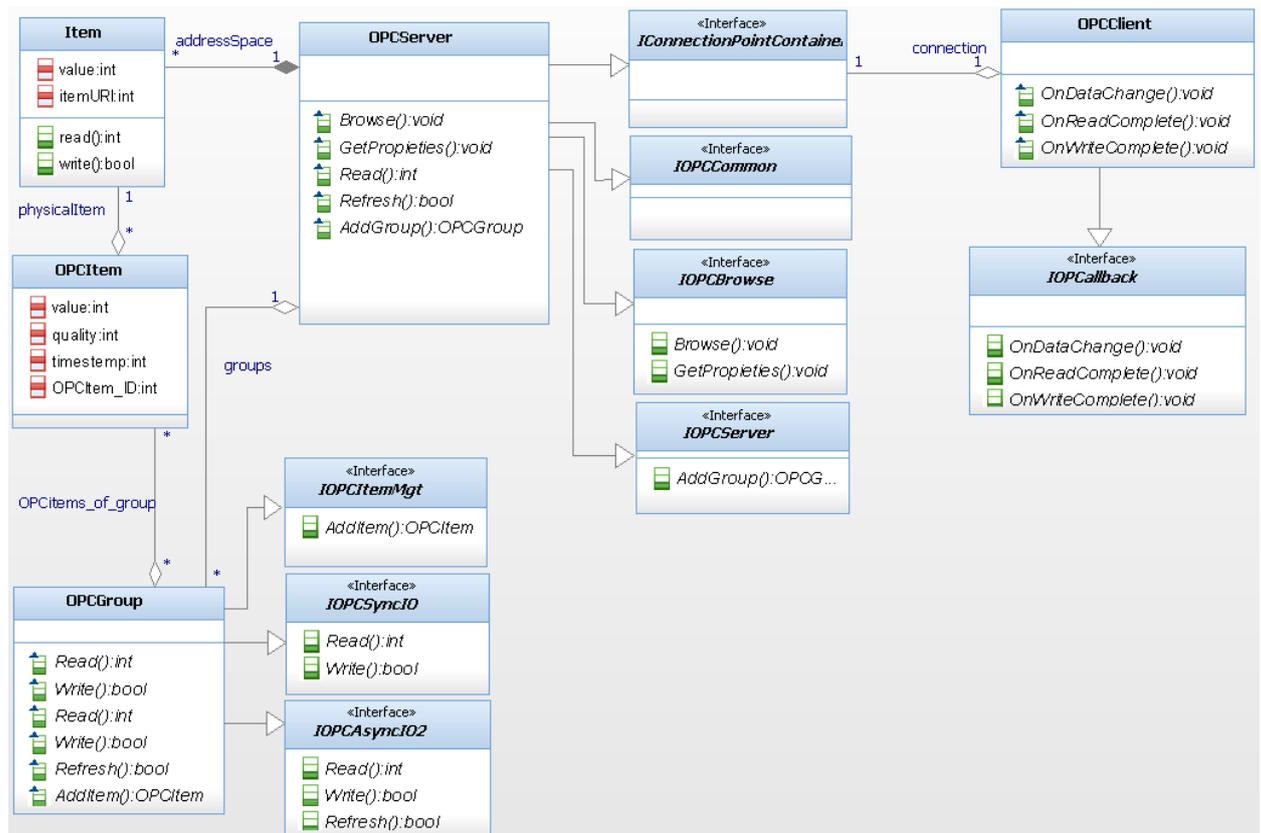


Abbildung 3 Klassendiagramm OPC Architektur

Betrachtet man einen OPC Server als ein Objekt beinhaltet dieses Objekt mehrere Softwareschnittstellen. Dasselbe gilt für jedes weitere OPC Objekt. Mittels dieser Schnittstellen können Objekte untereinander kommunizieren und Methoden untereinander aufrufen. Die eigentliche Kommunikation erfolgt über die Schnittstelle **IConnectionPointContainer** welche weiter unten (Kapitel: 2.2.4) erklärt wird. Der **OPCServer** besitzt einen Adressbaum in welchem alle URI's der angeschlossenen Prozessvariablen zu finden sind. Durch diese URI's hat man eine physikalische Verbindung zum entsprechenden Gerät.

2.2 OPC Item Objekt

Ein OPCItem Objekt zeichnet sich durch eine Verbindung zu einer Prozessvariable aus. Jede Prozessvariable beziehungsweise URI einer Prozessvariable befindet sich im Adressbaum des OPC Servers (Entspricht der Klasse Item in Abbildung 3 Klassendiagramm OPC Architektur). Jedes OPC Item wird durch eine einzigartige und vom Hersteller des OPC Servers festgelegte Item-ID festgelegt. Ein OPC Item Objekt besitzt folgende Eigenschaften:

- Wert
- Qualität
- Zeitstempel

Qualität zeigt ob der Wert zu einem gegebenen Zeitpunkt (Zeitstempel) korrekt ermittelt werden konnte. Die Qualität wird durch mehrere Eigenschaften beeinflusst, zum Beispiel ob eine Verbindung zur Prozessvariable beziehungsweise zum Gerät bestand. Der Zeitstempel definiert zu welcher Zeit der Wert aktualisiert wurde.

2.3 OPC Server Objekt

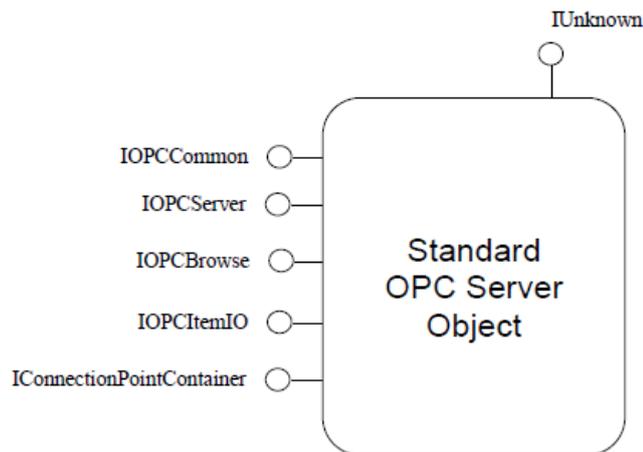


Abbildung 4 OPC Server Objekt

Die wichtigsten OPC Server Objekt Schnittstellen:

2.3.1 IUnknown

Ist eine von DCOM vorgeschriebene Schnittstelle.

2.3.2 IOPCCommon

Dient zum Teilen von Alarmen und Events zwischen zwei OPC Servern

2.3.3 IOPCServer

Ist das Hauptinterface eines OPC Servers. Es dient der Registration und Installation des OPC Servers innerhalb des Betriebssystems.

IOPCServer::AddGroup

Durch diese Methode kann vom OPCClient Objekt aus eine OPCGruppe (siehe: OPC Gruppen Objekt) auf dem OPCServer erstellt beziehungsweise instanziiert werden. Eine Gruppe ist ein Logischer Container der mehrere Zeiger auf OPCItems Objekte (siehe: OPC Item Objekt) beinhaltet. Beim Aufruf dieser Funktion wird dem OPCClient ein Zeiger (OPCGruppen Objekt Schnittstelle) zurückgegeben damit der OPCClient eine Referenz zur besagten OPCGruppe besitzt und mit dieser arbeiten kann.

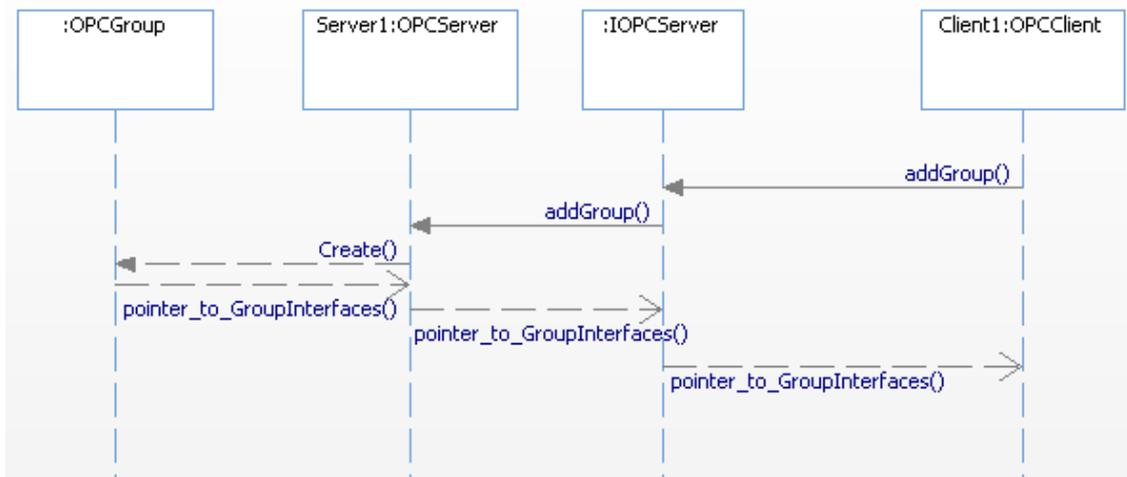


Abbildung 5 Sequenzdiagramm Gruppe hinzufügen

2.3.4 IConnectionPointContainer

Dient zur Herstellung einer Verbindung zum IOPCShutdown Interface innerhalb des Servers. Und zur Kommunikation zwischen zwei OPC Servern. Des Weiteren werden alle Schnittstellen dort gesammelt und falls ein Objekt eine Verbindung zu einer bestimmten Schnittstelle eines anderen Objekts haben möchte, kann über die besagte IConnectionPointContainer Schnittstelle Zugriff auf die gewünschte Schnittstelle erstellt werden. So ist auch die Kommunikation zwischen dem OPCClient und dem OPCServer beziehungsweise OPCGroup geregelt. Das genaue Vorgehen der Verbindungen unter Objekten kann in der DCOM Spezifikation gefunden werden.

2.3.5 IOPC Browse:

Dient als wichtige Schnittstelle um vom Client Objekt aus den Server Adressbaum zu durchsuchen. Ebenso können über diese Schnittstelle die Item Informationen abgefragt werden.

IOPCBrowse::Browse

Durch diese Methode kann ein Zweig oder der ganze Adressbaum auf dem OPCServer durchsucht werden. Die Ordner stellen die Netzwerke und Gruppen dar, die Dokumente die dazugehörigen Items. Man

übergibt dieser Methode die Server URI und erhält alle URI's in einer Hierarchie zurück. Befindet man sich z.B auf einer Gruppe und initialisiert ein Browse wird die URI der Gruppe der Browsemethode übergeben und als Ergebnis erhält man alle URI's der Unterelemente. Zu vergleichen mit der Hierarchie der Ordner unter Windows.



Abbildung 6 OPC Browse Hierarchie

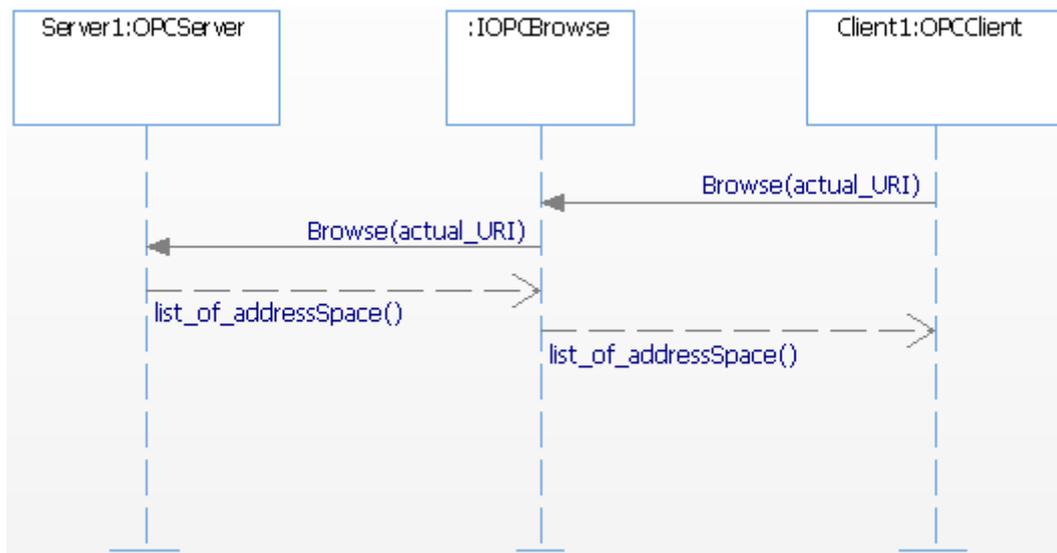


Abbildung 7 Sequenzdiagramm Browse Methode

2.4 OPC Client Objekt

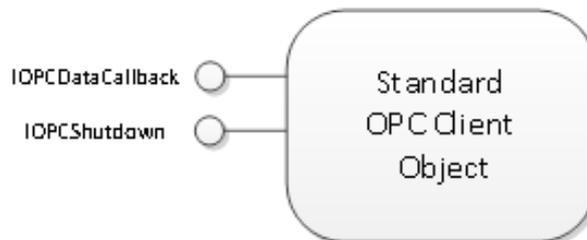


Abbildung 8 Standard OPC Client Objekt

Die wichtigsten OPC Client Schnittstellen sind:

2.4.1 IOPCDataCallback

Diese Schnittstelle wird benötigt um asynchrone lese und schreibe Operationen auf OPCItems des OPCServers auszuführen.

IOPCDataCallback::OnDataChanged

Diese Methode wird verwendet falls ein Client dem Server eine Refresh (IOPCAsyncIO2::Refresh2) Anfrage auf einer bestimmten Gruppe sendet. Dabei werden alle OPCItems der entsprechenden OPCGruppe kontrolliert und falls deren Werte geändert haben werden diese dem OPCClient über die besagte Methode mitgeteilt.

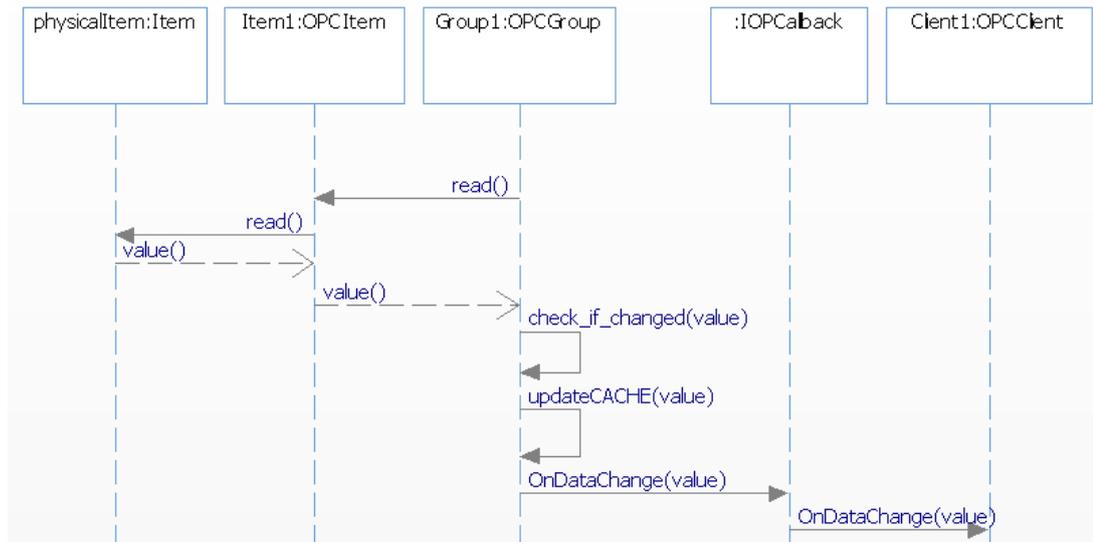


Abbildung 9 Sequenzdiagramm Items haben ihren Wert geändert

IOPCDataCallback::OnReadComplete

Diese Methode wird verwendet falls ein OPCClient einer OPCGruppe gewisse OPCItems abfragen möchte welche asynchron abgefragt werden. Dabei wird untersucht ob die OPCItems sich geändert haben, falls ja wird auf dem OPCServer der Cache aktualisiert und anschliessend via OnReadComplete Methode dem OPCClient mitgeteilt.

Siehe Abbildung 14 Sequenzdiagramm asynchrones lesen

IOPCDataCallback::OnWriteComplete

Diese Methode wird verwendet falls ein OPCClient ein oder mehrere OPCItems innerhalb einer OPCGruppe beschreiben möchte. Dabei sendet er via IOPCAsyncIO2 Schnittstelle welche OPCItems mit welchem Wert beschrieben werden sollen. Sobald die OPCGruppe die besagten OPCItems auf dem OPCServer beschrieben hat, wird dem OPCClient via IOPCDataCallback::OnWriteComplete Methode mitgeteilt dass die OPCItems beschrieben wurden.

Siehe Abbildung 15 Sequenzdiagramm asynchrones schreiben

2.5 OPC Gruppen Objekt

Das OPC Server Objekt besitzt falls die oben erwähnte Methode AddGroup aufgerufen wird ein OPC Gruppen Objekt. Dieses wiederum kann Zeiger zu mehreren OPCItems besitzen welche dem OPC Server Objekt gehören.

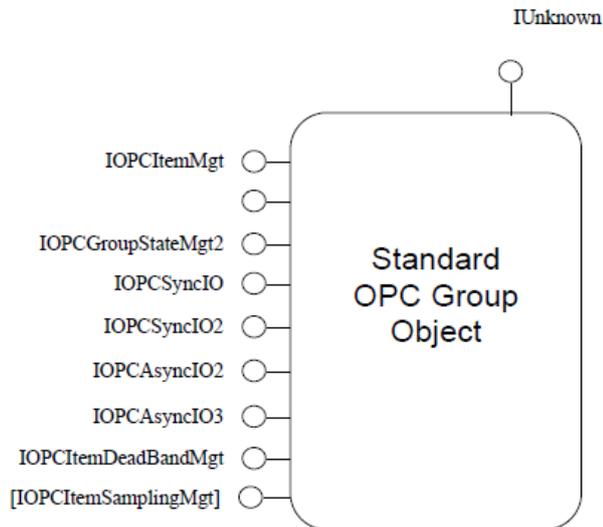


Abbildung 10 Standard OPC Gruppen Objekt

Wichtigsten Schnittstellen des OPC Gruppen Objekts:

2.5.1 IOPCItemMgt

Mittels dieser Schnittstelle kann vom OPCClient aus jedes OPCItem innerhalb dieser OPCGruppe verwaltet werden. Somit ist es möglich diverse OPCItems hinzuzufügen zu verändern und zu entfernen. Die genauen Methoden innerhalb dieser Schnittstelle sind folgende:

- IOPCItemMgt::AddItems
- IOPCItemMgt::ValidateItems
- IOPCItemMgt::RemoveItems
- IOPCItemMgt::SetActiveState
- IOPCItemMgt::SetClientHandles
- IOPCItemMgt::SetDatatypes
- IOPCItemMgt::CreateEnumerator

IOPCItemMgt::AddItems

Diese Methode erlaubt es vom OPCClient Objekt aus der OPCGruppe ein oder mehrere OPCItems hinzuzufügen. Dabei wird die Item URI übergeben um der OPCGruppe das entsprechende OPCItem mit seiner Verbindung zum Item hinzuzufügen.

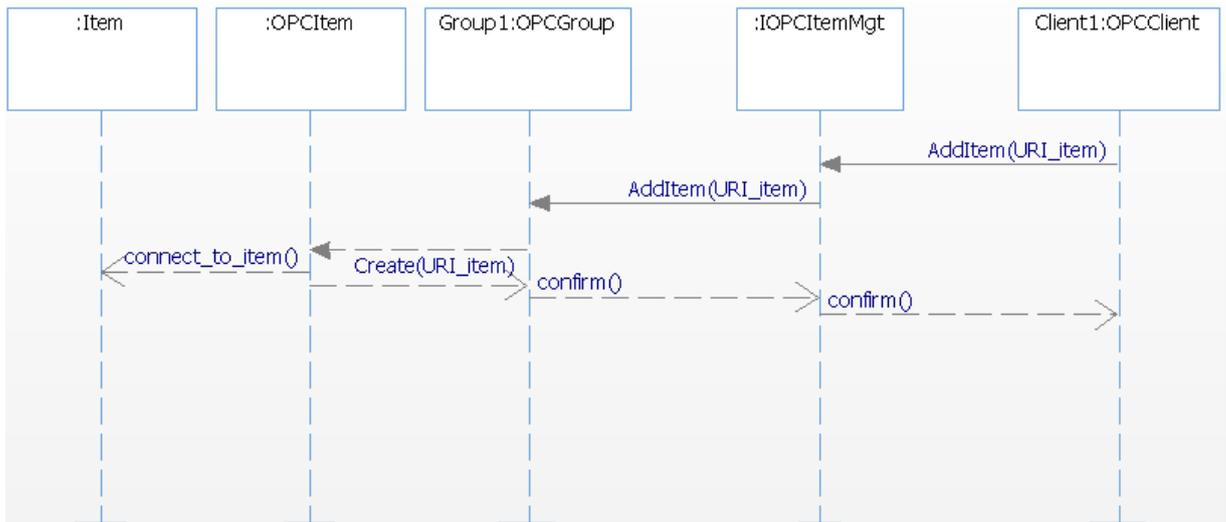


Abbildung 11 Sequenzdiagramm AddItem

2.5.2 IOPCGroupStateMgt

Über diese Schnittstelle können Parameter über das entsprechende OPCGruppen Objekt verändert werden. (Name, Status, usw.)

- IOPCGroupStateMgt::GetState
- IOPCGroupStateMgt::SetState
- IOPCGroupStateMgt::SetName
- IOPCGroupStateMgt::CloneGroup

2.5.3 IOPCSyncIO

Durch diese Schnittstelle können synchrone lese und schreibe Operationen auf dem OPCServer ausgeführt werden. Dabei stehen folgende Methoden zur Verfügung

IOPCSyncIO::Read

Diese Methode erlaubt es auf einem oder mehreren OPCItems die durch OPCItem_ID (Item 1 in diesem Fall) spezifiziert sind zu lesen. Dabei kann auf dem Gerät oder auf dem Cache des OPCServers gelesen werden.

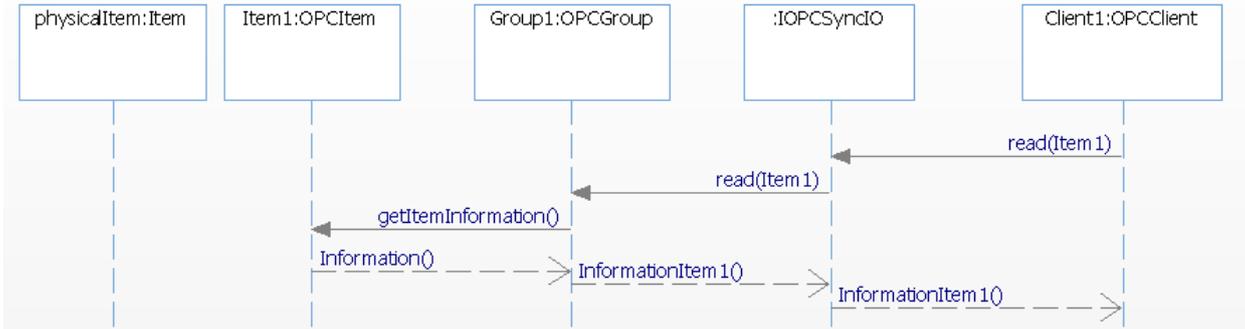


Abbildung 12 Sequenzdiagramm im Cache lesen

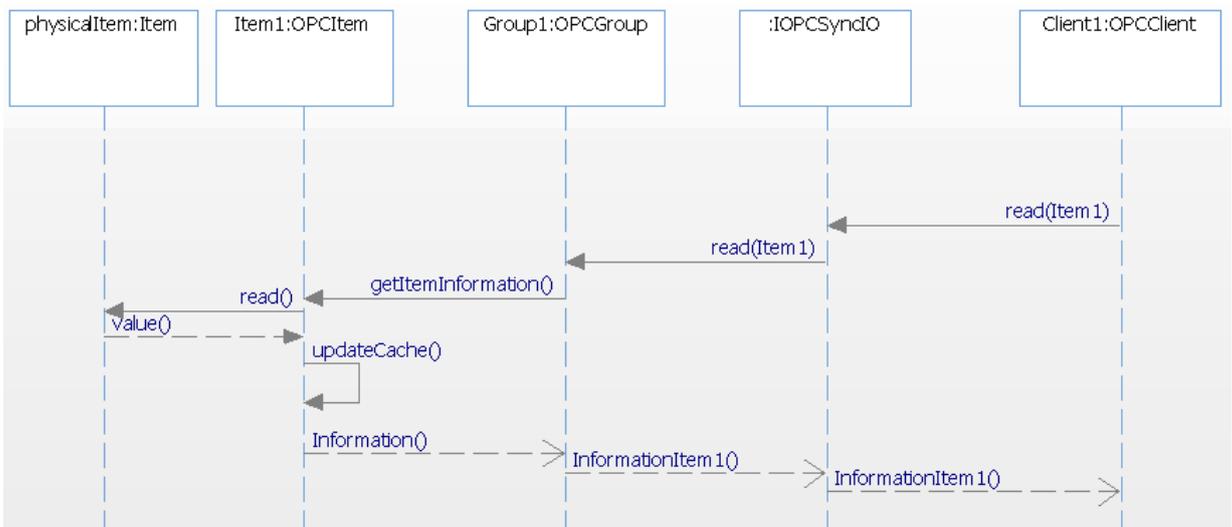


Abbildung 13 Sequenzdiagramm auf einem Gerät lesen

IOPCSyncIO::Write

Diese Methode erlaubt es auf einem oder mehreren OPCItems die durch OPCItem_ID (Item 1 in diesem Fall) spezifiziert sind zu schreiben. Dabei kann nur auf dem Gerät geschrieben werden dies erfolgt über die URI.

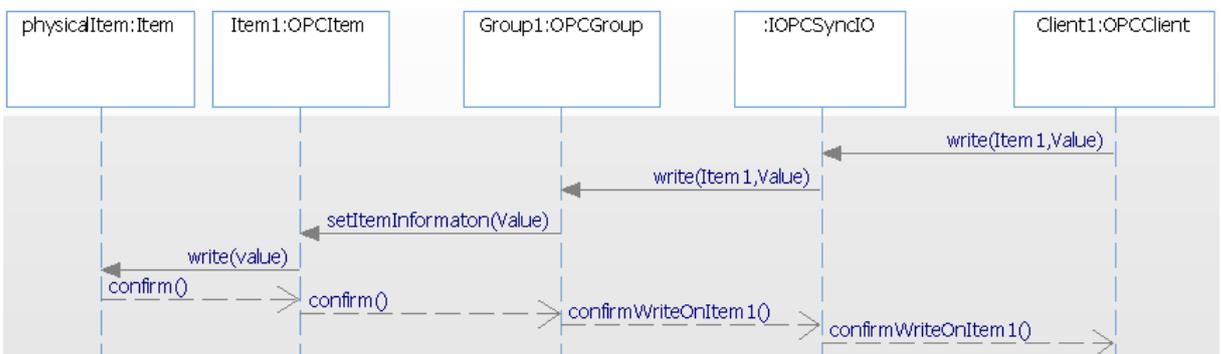


Abbildung 14 Sequenzdiagramm in ein Gerät schreiben

2.5.4 IOPCAsyncIO2

In dieser Schnittstelle können lese und schreibe Operationen auf dem OPCServer ausgeführt werden. Dabei werden die Anfragen in einer Liste gespeichert (Anfrage und TransaktionsID). Sobald die Anfrage gespeichert wurde wird dieser Prozess wieder freigegeben. Somit blockiert der OPCClient nicht die anderen Prozesse während dieser auf die Antwort des OPCServer wartet. Sobald die Anfrage vom OPCServer bearbeitet wurde sendet dieser das Resultat durch die IOPCDataCallback (siehe: IOPCDataCallback) Schnittstelle des Clients zurück.

IOPCAsyncIO2::Read

Diese Methode erlaubt das Lesen eines oder mehrerer OPCItems innerhalb einer OPCGruppe. Die Werte werden dem OPCClient mittels IOPCDataCallback Verbindung mitgeteilt. Diese Operation wird auf dem Gerät via URI ausgeführt.

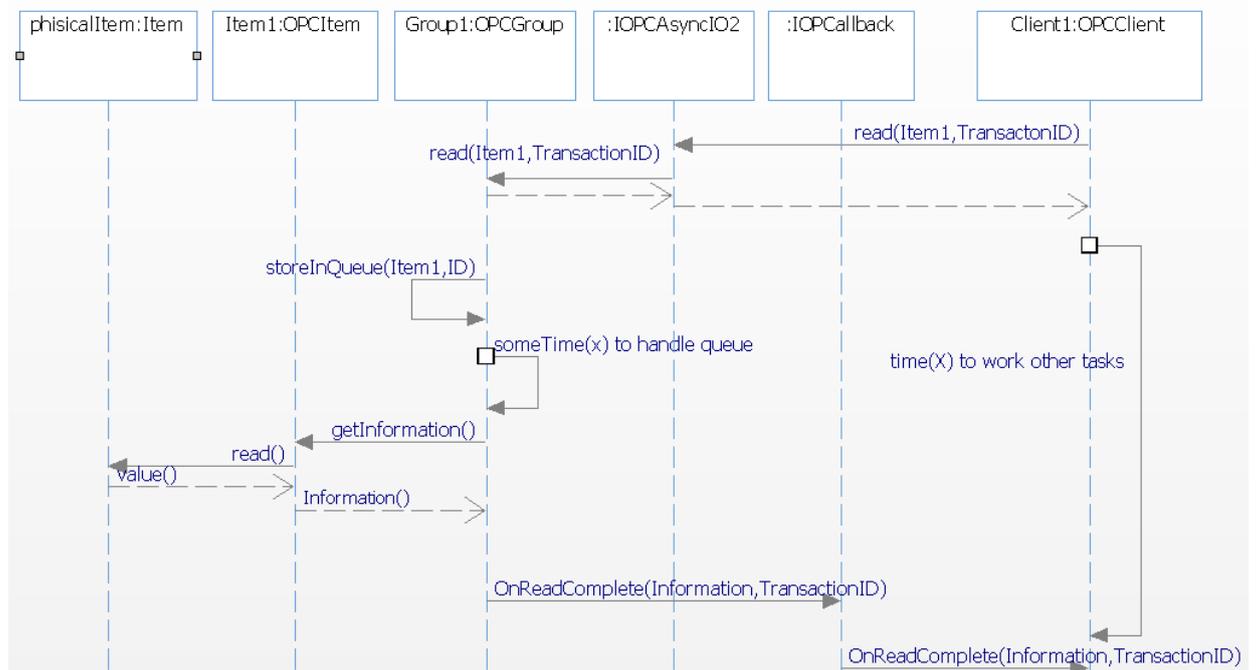


Abbildung 15 Sequenzdiagramm asynchrones lesen

IOPCAsyncIO2::Write

Diese Methode erlaubt das Schreiben eines oder mehrerer OPCItems innerhalb einer OPCGruppe über die URI des Items. Die Bestätigung folgt durch die IOPCDataCallback Verbindung.

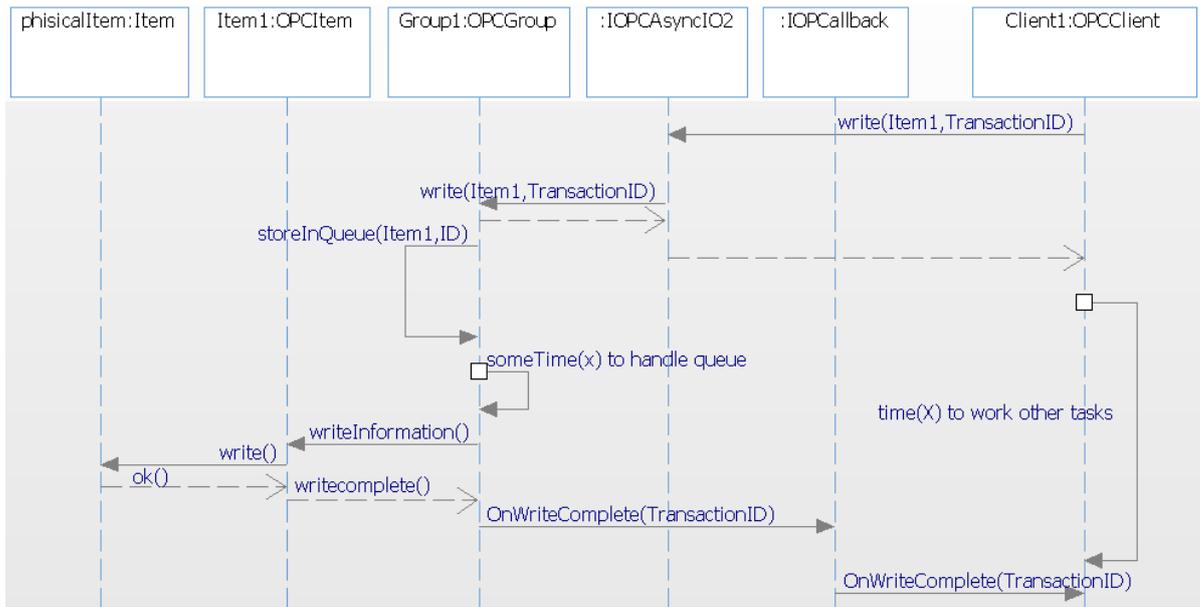


Abbildung 16 Sequenzdiagramm asynchrones schreiben

IOPCAsyncIO2::Refresh2

Diese Methode erzwingt die Aktualisierung aller betroffenen OPCItems innerhalb der OPCGruppe ihre Werte neu abzufragen und diese anschliessend durch die IOPCDataCallback::OnDataChange dem Client mitzuteilen. In diesem Fall wird nur ein OPCItem (Item 1) abgefragt.

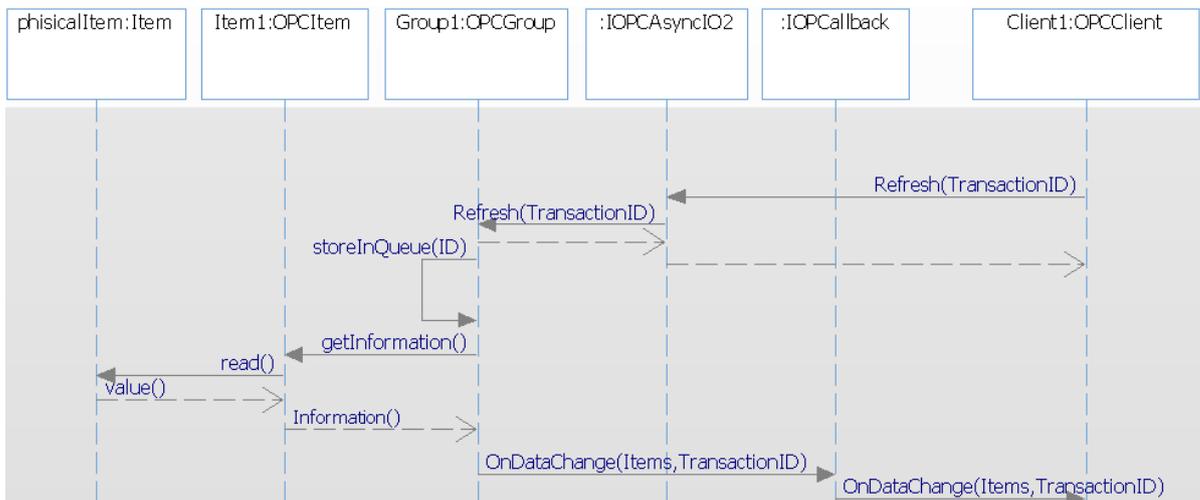


Abbildung 17 Sequenzdiagramm aktualisieren auf der Gruppe

3 EVALUATION

3.1 Einleitung

In Zusammenarbeit mit HYDRO-Exploitation soll die bestehende OPC Infrastruktur welche zurzeit von mehreren verschiedenen einzelnen OPC Servern besteht in nur einem OPC Server zentralisiert werden. Im Vordergrund steht die Kostenreduktion und das vereinfachte Verwalten der Prozesse beziehungsweise das Zentralisieren der verschiedenen Prozesse. Es werden 5 OPC Server Hersteller anhand einer Kriterienliste evaluiert welche in Zusammenarbeit mit HYDRO-Exploitation erarbeitet wurde. Die zu evaluierenden Hersteller sind:

- Triangle MicroWorks
- Kepware
- ReLab Software
- Matrikon
- Softing

3.2 Kriterienliste

Folgende Kriterien gilt es zu erfüllen:

- Zu unterstützende Protokolle:
 - Modbus TCP & Seriell
 - IEC 101 & IEC 104
 - 61850
 - Simens ProfiNET
- Modularität, Möglichkeit weitere Geräte direkt oder mit proprietärem Protokoll zu implementieren (Driver).
- Leichtigkeit zur Item Implementierung (automatisiert) z.B via Skript.
- Preis (Lizenzen, Item Limitation)
- Lizenz Verwaltung (Multilizenzen)
- Weg zu UA Falls DCOM nicht mehr unterstützt wird. (Verbindung von DA zu UA)
- Timestamp Funktion. Auf OPC Server Seite oder unterstütz t das angeschlossene Gerät diese Option in Zusammenarbeit mit dem OPC Server

3.3 Ergebnis

Grundsätzlich sieht man weiter unten, dass zwei der 5 evaluierten OPCServer (Kepware & Matrikon) eine gerätorientierte Ansicht besitzen und das Kommunikationsprotokoll für jedes Gerät einzeln in einem Plug-In implementiert wird.

Der Softing Hersteller implementiert das ganze standardisierte Protokoll einmal und nimmt an, dass die angeschlossenen Geräte mit nur diesem standardisierten Protokoll kommunizieren.

ReLab Software verfolgt die ähnliche Strategie wie es Softing tut. Dabei werden die Protokolle einmalig implementiert und die daran angeschlossenen Geräte an einen OPCServer gekoppelt.

Der Hersteller Triangle MicroWorks verfolgt eine ähnliche Strategie wie es ReLab Software bereits tut. Der wesentliche Unterschied liegt darin, dass die angeschlossenen Geräte nicht innerhalb eines OPCServer gekoppelt werden sondern in einer unveränderbaren Adressbaum. Somit lassen sich keine OPCGruppen darauf erstellen und verwalten.

3.3.1 Triangle MicroWorks



TRIANGLE MICROWORKS, INC.

Dieser Hersteller bietet keinen OPCServer im eigentlichen Sinne. Er verfolgt viel mehr die Strategie die Informationen aus den Protokollen

- Modbus TCP & Seriell
- IEC 101 & IEC 104
- 61850

Herauszufiltern und diese in einem Adressbaum abzuspeichern. Diese werden der OPC Schnittstelle zur Verfügung stellt. Somit kann bei Triangle MicroWorks nicht von einem OPCServer im eigentlichen Sinne gesprochen werden sondern von einem Protokoll Gateway.

3.3.2 ReLab Software



Dieser Hersteller bietet wie Triangle MicroWorks einen Protokoll Gateway zur Verfügung wobei ein OPCServer intern zur Verfügung steht. Dabei werden die Protokolle

- Modbus TCP & Seriell
- IEC 101 & IEC 104
- 61850

Unterstützt. Es können noch weitere Driver installiert werden.

3.3.3 Softing



Dieser Hersteller bietet einen Multiprotokoll OPCServer zur Verfügung. Die unterstützten Protokolle sind:

- Modbus TCP & Seriell
- Siemens ProfiNet

Einige weitere Protokolle werden noch angeboten.

3.3.4 Kepware



Dieser Hersteller bietet einen OPCServer welcher an seine angeschlossenen Geräte Orientiert ist. Dies bedeutet jedes Angeschlossene Gerät besitzt auf dem OPCServer einen spezifischen Treiber. Somit werden nicht die besagten Protokolle sondern die angeschlossenen Geräte mit spezifischen Treibern unterstützt.

Die Liste der Unterstützten Geräte mit den entsprechenden Plug-Ins lassen sich auf www.kepware.com finden.

3.3.5 Matrikon



Dieser Hersteller bietet einen OPC Server mit demselben Prinzip wie Kepware es tut. Jedes Angeschlossene Gerät besitzt einen Treiber falls dieses vom OPCServer Hersteller unterstützt wird.

Die Liste der Unterstützten Geräte mit den entsprechenden Plug-Ins lassen sich auf www.matrikon.com finden.

3.3.6 Zusammenfassung der Evaluation

	Triangle MicroWorks		Kepware	ReLab Software	Matrikon	Softing
	Master	Slave				
Modbus TCP & Seriell	yes	yes	device plugins	yes	device plugins	yes
IEC 101 & IEC 104	yes	yes		yes		no
61850	yes	yes		yes		no
Simens ProfiNet	no	no		no		yes
plugins for other protocolls or other devices	no	no	yes	yes (not a lot)	yes	yes (not a lot)
skript item implemenation	no	no	csv	deponds on protocoll	xml	xml
price	per year		per year	on demand	on demand	on demand
lizencetype	single pc or usb		driver per year	on demand	on demand	on demand
UA Support	no	no	yes	unknown	yes	yes
Timestamp function	yes	yes	yes	unknown	yes	yes

Abbildung 18 Zusammenfassung OPC Evaluation

4 SCHLUSSFOLGERUNG

Die Semesterprojektziele wurden erreicht. Das Einarbeiten in die OPC Schnittstelle wurde mittels Simulationsumgebung und der Analyse der OPC Schnittstellenspezifikation erreicht. Dabei wurde der OPCServer und seine Architektur in UML entworfen. Das Verhalten des OPCServers und des OPCClients wurde mittels Sequenzdiagramme aufgezeigt. Es wurde ein fundamentales Verständnis der Vorgänge innerhalb eines OPCServers geschaffen. Die Evaluation der OPCServer Hersteller wurde nach den vereinbarten Kriterien welche in der Sitzung vom 06.03.2013 (Siehe Anhang: Sitzungsprotokoll 06.03.2013) erarbeitet wurden durchgeführt. Das Ergebnis wurde vor den Verantwortlichen präsentiert. Das weitere Vorgehen wurde in der Sitzung vom 10.04.2013 (Siehe Anhang Sitzungsprotokoll 10.04.2013) definiert.

5 BIBLIOGRAPHY

- [1] O. T. Institute, «OPC & DCOM Diagnostics Hands-On OPC Training - Level 1,» OPC Training Institute (OPCTI), 2010.

6 ANHANG

- Sitzungsprotokoll vom 06.03.2013
- Sitzungsprotokoll vom 10.04.2013

Sitzungsprotokoll vom 06.03.2013

Begrüßung

Ausarbeitung der Kriterien liste

- Zu unterstützende Protokolle:
 - Modbus TCP & Seriell
 - IEC 101 & IEC 104
 - 61850
 - Simens ProfiNET
- Modularität, Möglichkeit weitere Geräte direkt oder mit proprietärem Protokoll zu implementieren (Driver).
- Leichtigkeit zur Item Implementierung (automatisiert) z.B via Skript.
- Preis (Lizenzen, Item Limitation)
- Lizenz Verwaltung (Multilizenzen)
- Weg zu UA Falls DCOM nicht mehr aktuell ist. (Verbindung von DA zu UA)
- Timestamp Funktion. Auf OPC Server Seite oder unterstütz t das angeschlossene Gerät diese Option in Zusammenarbeit mit dem OPC Server

Zu evaluierende Hersteller von OPC Servern

- Triangle MicroWorks
- Kepware
- ReLab Software
- Matrikon
- Softing

Zu erfüllende Aufgaben bis zur nächsten Sitzung:

- 3 halbe Tage fürs einarbeiten in die OPC Architektur
- 3 halbe Tage für die Evaluation der OPC Server der Hersteller anhand der Kriterien liste

Planung der nächsten Sitzung 10.04.2013 in der HES-SO Sion. Zeit muss noch bestätigt werden.

Sitzungsprotokoll vom 10.04.2013

Anwesende Personen

- Dominique Gabioud
- Maniero Gianantonio
- Pascal Mengis

Begrüßung

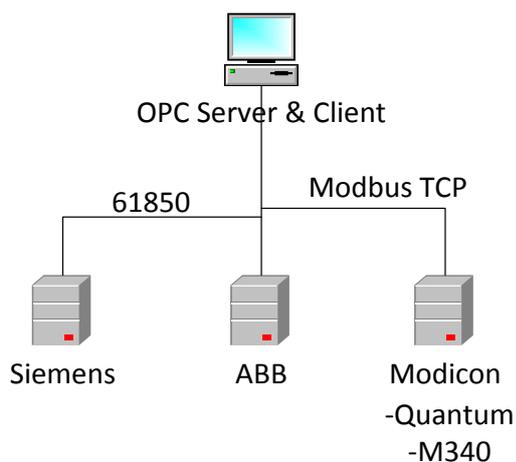
Präsentation der Evaluation

Präsentation der 5 evaluierten Herstellern von OPC Servern.

Besprechen des weiteren Vorgehens

Es gilt eine Infrastruktur zu erstellen welche 3 verschiedene OPC Server mit bestimmten Protokollen und Geräten kommunizieren lässt.

Die zu implementierende Infrastruktur



OPC Server welche ausgewählt wurden um in einer Testumgebung zu implementieren:

- Matrikon
- Kepware
- ReLab Software

Die benötigten Geräte Siemens, ABB und Modicon werden von der Schule beziehungsweise Hydro – Exploitation zur Verfügung gestellt.

In einem ersten Schritt soll die oben gezeigte Infrastruktur implementiert werden. Dabei werden noch keine Endgeräte an den Siemens, ABB und Modicon Geräten angeschlossen. Es gilt mit einer minimalen Konfiguration auf den besagten Geräten eine Kommunikation zwischen dem OPC Servern und den Geräten zu bewerkstelligen.

In einem zweiten Schritt sollen Endgeräte (Sensoren oder Schalter) an den Siemens,ABB und Modicon hinzugefügt werden und in den OPC Servern ersichtlich gemacht werden.

Zu erfüllende Aufgaben bis zur nächsten Sitzung:

Pascal Mengis:

- Anfrage an die OPC Server Herstellern verfassen um eine funktionstüchtige Version zu erhalten. Diese Anfrage an Herrn Gabioud senden.
- Infrastruktur in stand Setzen falls bereits eine OPC Server Version erhalten wurde.

Dominique Gabioud:

- Senden der von Pascal Mengis verfassten Anfragen an die betreffenden Hersteller.
- Nächsten Sitzungstermin in der Woche vom 06.05.2013 – 10.05.2013 bestimmen.

Anhang 2

LIST import.xlsx

Shared Name File Ver. 1.1
 ## File created by: GE FANUC -- CIMPPLICITY Ver: 7.50 Build 14261
 ## File created at: 15:30:43 on 05/24/2013.
 ##

PT_ID	ADDR	ADDR	ALM_CLASS	ALM_MSG	DESC	DEVICE_ID	PT_TYPE	RESOURCE_ID	UPDATE_CRITERIA
SAL_G2_EAG_AL_SURV_125V	40335	1	S2DC2	G2 EAG	Alarme manque de tension 125VDC	SAL_G2_EAG	BOOL	SAL_G2_EDC	UC
SAL_G2_EAG_AL_SURV_400V	40335	2	S2AC0	G2 EAG	Alarme manque de tension 400VAC	SAL_G2_EAG	BOOL	SAL_G2_EAC	UC
SAL_G2_EAG_AL_SURV_48V	40335	0	S2DC2	G2 EAG	Alarme manque de tension 48VDC	SAL_G2_EAG	BOOL	SAL_G2_EDC	UC
SAL_G2_EAG_AN	400211	15	S2AG8	G2 EAG	Séquence d'arrêt normal	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_AR	400212	15	S2AG8	G2 EAG	Séquence d'arrêt rapide	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_AU	400213	15	S2AG8	G2 EAG	Séquence d'arrêt d'urgence	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_DEM	400209	15	S2AG8	G2 EAG	Séquence de démarrage active	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_DISJ_H_AN	400210	4	S2AG8	G2 EAG	Ouverture du disjoncteur 13kV	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_DISJ_H_A_AR	400212	4	S2AG8	G2 EAG	Ouverture du disjoncteur 13kV	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_DISJ_H_F_EAU_AU	400213	1	S2AG8	G2 EAG	Fermeture hydraulique et ouverture Disjoncteur 13kV	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_EXC_EN_DEM	400209	1	S2AG8	G2 EAG	Enclenchement de l'excitation	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_EXC_EN_TMAXERR	40331	5	S2AG0	G2 EAG	Temps d'enclenchement de l'excitation écolé	SAL_G2_EAG	BOOL	SAL_G2_EAG	UC
SAL_G2_EAG_F_EAU_AR	400212	1	S2AG8	G2 EAG	Fermeture de la partie hydraulique	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_GR_OFF	400208	0	S2AG8	G2 EAG	Groupe à l'arrêt	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_GR_ON	400209	4	S2AG8	G2 EAG	Groupe en service	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_LM_H_DEM	400208	11	S2AG8	G2 EAG	Ouverture du déflecteur	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_LUBR_H_A_GEN	400211	2	S2AG8	G2 EAG	Déclenchement de la lubrification	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_MBSP_DISJ_GR_PV_H	400950	0			Heures de service du disjoncteur de groupe	SAL_G2_EAG	DINT	SAL_G2_EDS	OC
SAL_G2_EAG_MBSP_DISJ_GR_PV_OI	400954	0			Nombre d'enclenchements du disjoncteur de groupe	SAL_G2_EAG	DINT	SAL_G2_EDS	OC
SAL_G2_EAG_MBSP_MOT_ORDRE_1	400400	0			Mot de commande groupe	SAL_G2_EAG	UINT	SAL_G2_EAG	OC
SAL_G2_EAG_MBSP_MOT_ORDRE_2	400401	0			Mot de commande cpt 1	SAL_G2_EAG	UINT	SAL_G2_EAG	OC
SAL_G2_EAG_MBSP_MOT_ORDRE_3	400402	0			Mot de commande cpt 2	SAL_G2_EAG	UINT	SAL_G2_EAG	OC
SAL_G2_EAG_MBSP_MOT_ORDRE_4	400403	0			Mot de commande MER-MGP	SAL_G2_EAG	UINT	SAL_G2_EAG	OC
SAL_G2_EAG_MBSP_PAR_1	400101	0			Alarme Température enroulements statoriques 1 à SAL_G2_EAG	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_10	400110	0			Alarme Température eau réfrigération entrée	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_11	400111	0			Alarme Température réfrigération sortie Alternateur	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_12	400112	0			Alarme Température réfrigération sortie Lubrificateur	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_16	400116	0			Déclenchement Temp. enroulements statoriques 1 SAL_G2_EAG	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_17	400117	0			Déclenchement Température flux d'air frais 1 et 2	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_18	400118	0			Déclenchement Température flux d'air chaud 1 et 2	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_19	400119	0			Déclenchement Température circuits magnétiques	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_2	400102	0			Alarme Température flux d'air frais 1 et 2	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_20	400120	0			Déclenchement Température palier turbine	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_21	400121	0			Déclenchement Température palier excitation	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_22	400122	0			Déclenchement Température huile de régulation	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_23	400123	0			Déclenchement Temp. lubrification avant réfrigéra	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_24	400124	0			Déclenchement Temp. lubrification après réfrigéra	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_25	400125	0			Déclenchement Température eau réfrigération ent	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_26	400126	0			Déclenchement Temp. réfrigération sortie Alternat	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_27	400127	0			Déclenchement Temp. réfrigération sortie Lubrific	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_3	400103	0			Alarme Température flux d'air chaud 1 et 2	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_31	400131	0			Temps de surveillance pour circul. huile de lubrifica	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_32	400132	0			Temps de surveillance pour circul. eau de réfrigéra	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_MBSP_PAR_33	400133	0			Temps de surveillance pression huile de régulation	SAL_G2_EAG	INT	SAL_G2_EAG	OS

SAL_G2_EAG_M BSP_PAR_34	400134	0	G2 Désactivation de la pompe de service de régulation	Temps de surveillance pression huile agrégat	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_4	400104	0	G2 Synchro Dead Bus autorisée	Alarme Température circuits magnétiques 1 à 3	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_45	400195	0	G2 Séquentiel en mode pas à pas	Seuil pour pression huile régulation OK	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_46	400146	0	G2 Séquentiel en mode pas à pas	Désactivation de la pompe de service de régulation	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_PAR_47	400146	1	G2 Séquentiel en mode pas à pas	Synchro Dead Bus autorisée	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_PAR_48	400146	2	G2 Séquentiel en mode pas à pas	Séquentiel en mode pas à pas	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_PAR_5	400105	0		Alarme Température palier turbine	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_6	400106	0		Alarme Température palier excitation	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_60	400146	14	G2 Activation Test délestages (survitesse)	Activation Test délestages (survitesse)	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_PAR_7	400107	0		Alarme Température huile de régulation	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_8	400108	0		Alarme Température lubrification avant réfrigérant	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PAR_9	400109	0		Alarme Température lubrification après réfrigérant	SAL_G2_EAG	INT	SAL_G2_EAG	OS
SAL_G2_EAG_M BSP_PMP1_MVS_PV	400918	0		Heures de fonctionnement de la pompe 1	SAL_G2_EAG	DINT	SAL_G2_MVS	OC
SAL_G2_EAG_M BSP_PMP1_MVS_PV	400926	0		Nombre d'enclenchements de la pompe 1	SAL_G2_EAG	DINT	SAL_G2_MVS	OC
SAL_G2_EAG_M BSP_PMP2_MVS_PV	400930	0		Heures de fonctionnement de la pompe 2	SAL_G2_EAG	DINT	SAL_G2_MVS	OC
SAL_G2_EAG_M BSP_PMP2_MVS_PV	400930	0		Nombre d'enclenchements de la pompe 2	SAL_G2_EAG	DINT	SAL_G2_MVS	OC
SAL_G2_EAG_M BSP_PMP_DEM_PV	400902	0		Heures de fonctionnement de la pompe de démarrage	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_PMP_DEM_PV	400906	0		Nombre d'enclenchements de la pompe de démarrage	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_PMP_DEM_PV	400934	0		Heures de fonctionnement de la pompe de lubrification	SAL_G2_EAG	DINT	SAL_G2_MGP	OC
SAL_G2_EAG_M BSP_PMP_DEM_PV	400938	0		Nombre d'enclenchements de la pompe de lubrification	SAL_G2_EAG	DINT	SAL_G2_MGP	OC
SAL_G2_EAG_M BSP_PMP_REFR_PV	400942	0		Heures de fonctionnement de la pompe de réfrigérant	SAL_G2_EAG	DINT	SAL_G2_MER	OC
SAL_G2_EAG_M BSP_PMP_REFR_PV	400946	0		Nombre d'enclenchements de la pompe de réfrigérant	SAL_G2_EAG	DINT	SAL_G2_MER	OC
SAL_G2_EAG_M BSP_PMP_SERV_PV	400910	0		Heures de fonctionnement de la pompe de service	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_PMP_SERV_PV	400914	0		Nombre d'enclenchements de la pompe de service	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_PPE_R_HUILLE_F	400958	0		Heures de fonctionnement de la pompe de récupération	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_PPE_R_HUILLE_F	400962	0		Nombre d'enclenchements de la pompe de récupération	SAL_G2_EAG	DINT	SAL_G2_MTU	OC
SAL_G2_EAG_M BSP_P_MOINS	400400	6	G2 EAG P -	P -	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_P_PLUS	400400	7	G2 EAG P +	P +	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_Q_MOINS	400400	0	G2 EAG Q -	Q -	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_M BSP_Q_PLUS	400400	1	G2 EAG Q +	Q +	SAL_G2_EAG	BOOL	SAL_G2_EAG	OC
SAL_G2_EAG_PRSP_ACC_A_DECL	40305	0	G2 MVS Accumulateur seuil A déclenchement	Accumulateur seuil A décl. pompe	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_ACC_B_1_ENCL	40305	1	G2 MVS Accumulateur seuil B Encl. 1 pompe	Accumulateur seuil B Encl. 1 pompe	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_ACC_C_2_ENCL	40305	2	G2 MVS Accumulateur seuil C Enclenchement 2	Accumulateur seuil C Encl. 2 pompes	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_ACC_D_DECL	40305	10	G2 MVS Accumulateur seuil D Déclenchement	Accumulateur seuil D Décl. group	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_ALIM_RES_EN	400206	5	G2 ERT Alimentation par réseau	Alimentation par réseau En	SAL_G2_EAG	BOOL	SAL_G2_ERT	OC
SAL_G2_EAG_PRSP_ALIM_SOUT_EN	400206	4	G2 ERT Alimentation par soutirage	Alimentation par soutirage En	SAL_G2_EAG	BOOL	SAL_G2_ERT	OC
SAL_G2_EAG_PRSP_AL_EAU_HUILLE	40310	13	G2 MTU Alarme détecteur eau dans huile	Alarme détecteur eau dans huile	SAL_G2_EAG	BOOL	SAL_G2_MTU	UC
SAL_G2_EAG_PRSP_AL_FAH_BOUCH	40310	15	G2 MTU Alarme filtre à huile bouché	Alarme filtre à huile bouché	SAL_G2_EAG	BOOL	SAL_G2_MTU	UC
SAL_G2_EAG_PRSP_AL_FILT_P	40305	7	G2 MVS Alarme filtre de pression	Alarme filtre de pression	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_AL_PMP1_2_FII	40310	9	G2 MTU Alarme pompe 1 ou 2 filtre bouché	Alarme pompe 1 ou 2 filtre bouché	SAL_G2_EAG	BOOL	SAL_G2_MTU	UC
SAL_G2_EAG_PRSP_AL_PRESS_EAU	40310	14	G2 MTU Alarme pression d'eau	Alarme pression d'eau	SAL_G2_EAG	BOOL	SAL_G2_MTU	UC
SAL_G2_EAG_PRSP_AR_EAU	40305	6	G2 MVS Alarme température	Alarme température	SAL_G2_EAG	BOOL	SAL_G2_MVS	UC
SAL_G2_EAG_PRSP_ASS_OUV	400204	9	G2 MTU Arrêt rapide par Matrice EAU (Généralités)	Arrêt rapide par Matrice EAU (Généralités)	SAL_G2_EAG	BOOL	SAL_G2_MTU	OC
SAL_G2_EAG_PRSP_ASS_P	400204	10	G2 MTU Asservissement puissance	Asservissement puissance	SAL_G2_EAG	BOOL	SAL_G2_MTU	OC
SAL_G2_EAG_PRSP_ASS_P_EN	400204	15	G2 MTU Asservissement puissance en service	Asservissement puissance en service	SAL_G2_EAG	BOOL	SAL_G2_MTU	OC
SAL_G2_EAG_PRSP_BLOC_CO2	40301	1	G2 EDI Blocage conduite CO2	Blocage conduite CO2	SAL_G2_EAG	BOOL	SAL_G2 EDI	UC
SAL_G2_EAG_PRSP_BLOC_MFC_CO	40301	2	G2 EDI Blocage mécanique CO2	Blocage mécanique CO2	SAL_G2_EAG	BOOL	SAL_G2 EDI	UC
SAL_G2_EAG_PRSP_CC_MAX	400204	12	G2 MTU Consigne de charge position max.	Consigne de charge position max.	SAL_G2_EAG	BOOL	SAL_G2_MTU	OC
SAL_G2_EAG_PRSP_CC_MIN	400204	11	G2 MTU Consigne de charge position min.	Consigne de charge position min.	SAL_G2_EAG	BOOL	SAL_G2_MTU	OC

Anhang 3

framework.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.hydro.ch"
  xmlns="http://www.hydro.ch"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  >

<!-- HIERARCHY -->
<xs:element name="HYDRO_P" >
  <xs:complexType>
    <xs:sequence>

      <xs:element name="scada" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="group" minOccurs="0" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element name="field" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="agent" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- SCADA PART OF HYDRO_P -->

<!-- group defines a OPCGroup from the OPC Client -->
<xs:element name="group">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="node" minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="name" use="required"/>
  </xs:complexType>
</xs:element>

<!-- NODE defines a folder in a group element -->
<xs:element name="node">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="m" minOccurs="0" maxOccurs="1" />
      <xs:element ref="p" minOccurs="0" maxOccurs="1" />
      <xs:element ref="s" minOccurs="0" maxOccurs="1" />
      <xs:element ref="a" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="name" use="required"/>
  </xs:complexType>
</xs:element>

<!-- alarm folder -->
<xs:element name="a">
  <xs:complexType>
    <xs:sequence>
```

```
<xs:element ref="point" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- signalisation folder -->
<xs:element name="s">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="point" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- parameter folder -->
<xs:element name="p">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="point" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- measurment folder -->
<xs:element name="m">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="point" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--
  POINT defines the OPCItems which are referenced to I/O
  Registers in the FIELD part of HYDRO_P
-->
<xs:element name="point">
  <xs:complexType>
    <xs:attribute name="description" use="required" />
    <xs:attribute name="datatype" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="DINT" />
          <xs:enumeration value="INT" />
          <xs:enumeration value="REAL" />
          <xs:enumeration value="SINT" />
          <xs:enumeration value="UDINT" />
          <xs:enumeration value="USINT" />
          <xs:enumeration value="BOOL" />
          <xs:enumeration value="BYTE" />
          <xs:enumeration value="WORD" />
          <xs:enumeration value="DWORD" />
          <xs:enumeration value="STRING" />
          <xs:enumeration value="STRING_8" />
          <xs:enumeration value="STRING_20" />
          <xs:enumeration value="STRING_80" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute ref="semantic" />
    <xs:attribute name="ref" use="required" type="xs:IDREF" />
    <xs:attribute name="alarm_msg" type="xs:string" use="optional" />
    <xs:attribute name="alarm_class" use="optional" >
      <xs:simpleType>
        <xs:restriction base="xs:string" >
```

```

        <xs:maxLength value="4" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>

</xs:element>

<!-- FIELD-->

<!-- agent is used to define how devices are attached (Driver & Ports...) -->
<xs:element name="agent">
    <xs:complexType>

        <xs:sequence>
            <xs:element ref="device" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
        <xs:attribute name="id" use="required" type="xs:integer" />
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="driver" use="required">
            <!-- verfügbare drivers -->
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="Modbus Ethernet" />
                    <xs:enumeration value="Modbus Serial" />
                    <xs:enumeration value="Siemens TCP/IP Ethernet" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="port" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="Ethernet" />
                    <xs:enumeration value="Serial" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="portnb" use="optional" />
    </xs:complexType>

</xs:element>

<!-- device is used to define all elementary elements of an device -->
<xs:element name="device">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="address" />
            <xs:element ref="items" />
        </xs:sequence>
        <xs:attribute name="id" use="required" type="xs:integer" />
        <xs:attribute name="name" type="xs:string" use="required" />
        <xs:attribute name="devicetype" use="required">
            <!-- verfügbare PLC Typen -->
            <xs:simpleType >
                <xs:restriction base="xs:string">
                    <!-- siemens Ethernet TCP/IP Devices -->
                    <xs:enumeration value="S7-200" />
                    <xs:enumeration value="S7-300" />
                    <xs:enumeration value="S7-400" />
                    <xs:enumeration value="S7-1200" />
                    <xs:enumeration value="NetLink: S7-300" />
                    <xs:enumeration value="NetLink: S7-400" />
                    <!-- Modbus Serial -->
                    <xs:enumeration value="Modbus RTU Device" />
                </xs:restriction>
            </xs:simpleType >
        </xs:attribute>
    </xs:complexType>

```

```
<!-- Modbus Ethernet -->
  <xs:enumeration value="Modbus Eth Device" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="location" use="optional"/>
</xs:complexType>
</xs:element>

<!-- address is used to define address of device -->
<xs:element name="address">
  <xs:complexType>
    <xs:sequence >
      <xs:element name="modbus">
        <xs:complexType>
          <xs:attribute name="address">
            <xs:simpleType>
              <xs:restriction base="xs:unsignedInt" >
                <xs:maxInclusive value="255"/>
                <xs:minInclusive value="0"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
      <xs:element name="ip">
        <xs:complexType>
          <xs:attribute name="address" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- items is used to define the items in the corresponding device -->
<xs:element name="items">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="collection" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="item" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- collection is used to define a undergroup of items on a device -->
<xs:element name="collection">
  <xs:complexType>
    <xs:sequence >
      <xs:element ref="item" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" use="required"/>
  </xs:complexType>
</xs:element>

<!-- item defines the I/O's of an device -->
<xs:element name="item">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="description"/>
      <xs:element name="address"/>
    </xs:sequence>
    <xs:attribute name="id" use="required" type="xs:ID" id="itemID"/>
  </xs:complexType>
</xs:element>
```

```

<xs:attribute name="name" type="xs:string" use="required" />
<xs:attribute name="datatype" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse" />
      <xs:enumeration value="Default" />
      <xs:enumeration value="Boolean" />
      <xs:enumeration value="Char" />
      <xs:enumeration value="Byte" />
      <xs:enumeration value="Short" />
      <xs:enumeration value="Word" />
      <xs:enumeration value="Long" />
      <xs:enumeration value="DWord" />
      <xs:enumeration value="Float" />
      <xs:enumeration value="Double" />
      <xs:enumeration value="String" />
      <xs:enumeration value="BCD" />
      <xs:enumeration value="LBCD" />
      <xs:enumeration value="Date" />
      <xs:enumeration value="Boolean Array" />
      <xs:enumeration value="Char Array" />
      <xs:enumeration value="Byte Array" />
      <xs:enumeration value="Short Array" />
      <xs:enumeration value="Word Array" />
      <xs:enumeration value="Long Array" />
      <xs:enumeration value="DWord Array" />
      <xs:enumeration value="Float Array" />
      <xs:enumeration value="Double Array" />
      <xs:enumeration value="String Array" />
      <xs:enumeration value="BCD Array" />
      <xs:enumeration value="LBCD Array" />
      <xs:enumeration value="Date Array" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute ref="semantic" use="optional" />
<xs:attribute name="rights" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse" />
      <xs:enumeration value="Read Only" />
      <xs:enumeration value="Read/Write" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

<!-- attribute semantic makes a pre set of the <item> and the <point> tags -->
<!-- add some new semantic for preset of the <item> and <point> tags -->
<xs:attribute name="semantic">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="speed" />
      <xs:enumeration value="speed_ind" />
      <xs:enumeration value="distance" />
      <xs:enumeration value="distance_ind" />
      <xs:enumeration value="over voltage" />
      <xs:enumeration value="button_ind" />
      <xs:enumeration value="pressure" />
      <xs:enumeration value="blocked" />
      <xs:enumeration value="pressure" />
      <xs:enumeration value="contact" />
      <xs:enumeration value="position min" />
    </xs:restriction>
  </xs:simpleType>

```

```
<xs:enumeration value="position max" />
<xs:enumeration value="close_ind" />
<xs:enumeration value="open_ind" />
<xs:enumeration value="on/off" />
<xs:enumeration value="temperature" />
<xs:enumeration value="command" />
<xs:enumeration value="level" />
</xs:restriction>
</xs:simpleType>
</xs:attribute>

</xs:schema>
```

Anhang 4

Gr3 Liste superviseur.xls

Miéville Gr2 : Liste des signaux pour Superviseur (sans Horodatage)

Adresse	ADD SCA	MSTR	ADD MSTR	Bit	Libelle	Designation	Classe
%MW 200	400200	1		0	TCSP_CMD_DIST	EAG Commande en mode Distance	7
				1	TCSP_CMD_AUTO	EAG Commande en mode Local automatique	7
				2	TCSP_CMD_MAN	EAG Commande en mode Local manuel	7
				3	TCSP_CMD_REV	EAG Commande en mode Révision	7
				4	TCSP_SYNC_AUTO	Synchronisation Automatique	7
				5	TCSP_SYNC_HORS	Synchronisation Hors	7
				6	TCSP_SYNC_MAN	Synchronisation Manuelle	7
				7	S16_CMD_CEG	Autorité de commande CEG	7
				8	S16_CMD_MIEV	Autorité de commande Miéville	7
				9	TCSP_MVS_AUTO	MVS Commande en mode Automatique	7
				10	TCSP_MVS_MAN	MVS Commande en mode Manuel	7
				11	TCSP_MVS_REV	MVS Commande en mode Révision	7
				12	TCSP_MGP_AUTO	MGP Commande en mode Automatique	7
				13	TCSP_MGP_MAN	MGP Commande en mode Manuel	7
				14	TCSP_MGP_REV	MGP Commande en mode Révision	7
%MW 201	400201	1		0	TCSP_CMD_DISJ_HORS	Commande Disjoncteur 13 kV Hors	7
				1	TCSP_CMD_DISJ_EN	Commande Disjoncteur 13 kV En	7
				2	TCSP_CMD_EXC_EN	Commande Excitation En / Hors	7
				3	TCSP_CMD_EXC_MAN	Commande Excitation Manuel	7
				4	TCSP_CMD_EXC_AUTO	Commande Excitation Auto	7
				5	TCSP_EXC_MOINS	Commande Excitation -	7
				6	TCSP_EXC_PLUS	Commande Excitation +	7
				7	TCSP_EXE_CMD	EAG Bouton d'exécution	7
				8			
				9			
				10			
				11			
				12			
				13			
				14	TCSP_SYNC_60KV_AUTO	Synchronisation 60KV mode auto	7
15	TCSP_SYNC_60KV_HORS	Synchronisation 60KV mode hors	7				
%MW 202	400202	2		0	TCSP_MTU_PMP_LUB	Pompe à huile du palier En / Hors	7
				1	TCSP_PMP_DEM_REG_CMD	Pompe démarrage régulateur de turbine En / Hors	7
				2	TCSP_MTU_PMP_REF	Pompe à eau réfrigération En/ Hors	7
				3	TCSP_SCD_CMD	Défecteur En / Hors	7
				4	TCSP_SCP_CMD	Pointeau En / Hors	7
				5	TCSP_LIM_MEC_CMD_F	Ordre de fermeture Limiteur mécanique	7
				6	TCSP_LIM_MEC_CMD_O	Ordre d'ouverture Limiteur mécanique	7
				7	TCSP_TU_EN_CMD	Turbine En	7
				8	TCSP_TU_HORS_CMD	Turbine Hors	7
				9	TCSP_CYL_PLUS	Limiteur électrique +	7
				10	TCSP_CYL_MOINS	Limiteur électrique -	7
				11	TCSP_CN_PLUS	Consigne de vitesse +	7
				12	TCSP_CN_MOINS	Consigne de vitesse -	7
				13	TCSP_CC_PLUS	Consigne de charge +	7
				14	TCSP_CC_MOINS	Consigne de charge -	7
15	TCSP_MTU_EXE_CMD	MTU Bouton d'exécution	7				
%MW 203	400203	2		0	TCSP_MVS_EXE_CMD	MVS Bouton d'exécution	7
				1	TCSP_MVS_PMP1_CMD	Pompe 1 En / Hors MVS	7
				2	TCSP_MVS_PMP2_CMD	Pompe 2 En / Hors MVS	7
				3	TCSP_VANNE_CMD_O	Ordre ouverture de la vanne sphérique	7
				4	TCSP_VANNE_CMD_F	Ordre fermeture de la vanne sphérique	7
				5			
				6			
				7			
				8	TCSP_MGP_EXE_CMD	MGP Bouton d'exécution	7
				9	TCSP_MGP_PMP_LUB	Pompe Lubrification En / Hors MGP	7
				10	TCSP_MGP_PMP_REF	Pompe Réfrigération En / Hors MGP	7
				11			
				12			
				13			
				14			
15							
%MW 204	400204	3		0	PRSP_SCD_FERM	Défecteur fermé	7
				1	PRSP_SCD_OUV	Défecteur ouvert	7
				2	PRSP_SCP_FERM	Pointeau fermé	7
				3	PRSP_SCP_M_VIDE	Pointeau position marche à vide	7
				4	PRSP_LIM_MEC_FERM	Limiteur mécanique fermé	7
				5	PRSP_LIM_MEC_OUV	Limiteur mécanique ouvert	7
				6			
				7			
				8	PRSP_RES_ISOLE	Réseau isolé	7
				9	PRSP_ASS_OUV	Asservissement ouverture	7
				10	PRSP_ASS_P	Asservissement puissance	7
				11	PRSP_CC_MIN	Consigne de charge position min.	7
				12	PRSP_CC_MAX	Consigne de charge position max.	7
				13	PRSP_LIM_EL_MIN	Limiteur électrique position min.	7
				14	PRSP_LIM_EL_MAX	Limiteur électrique position max.	7
15	PRSP_ASS_P_EN	Asservissement puissance en service	7				
%MW 205	400205	3		0	PRSP_P_EAU_AM	MVS Pression eau amont présente	7
				1	PRSP_P_EAU_AV	MVS Pression eau aval présente	7
				2	PRSP_M_A_HYDR_AUTO	MVS Interrupteur hydraulique en mode Automatique	7
				3	PRSP_J_REV	MVS Joint de révision En	7
				4	SPPR_CMD_J_SERV	MVS Commande ouverture joint de service	7
				5	SPPR_CMD_BYP	MVS Commande ouverture Bypass	7
				6	SPPR_VANNE_CMD_O_F	MVS Commande ouverture Vanne	7
				7			
				8	PRSP_VANNE_POS_O	MVS Position Vanne Ouverte	7
				9	PRSP_VANNE_POS_F	MVS Position Vanne Fermée	7
				10			
				11			
				12			
				13			
				14			
15							

EAG

EAG

EAG

MVS

MGP

MTU

MVS

Adresse	ADD SCA	MSTR	ADD MSTR	Bit	Libelle	Designation	Classe
%MW 206	400206	4		0	PRSP DIOD_OUV	Régulateur de tension Diode ouverte	2
				1			
				2	PRSP_CONS_MAX	Régulateur de tension Consigne Max	7
				3	PRSP_CONS_MIN	Régulateur de tension Consigne Min	7
				4	PRSP ALIM_SOUT_EN	Régulateur de tension Alimentation par Soutirage En	7
				5	PRSP ALIM_RES_EN	Régulateur de tension Alimentation par Réseau En	7
				6	PRSP_REG_U_AUTO	Régulateur de tension en mode Automatique	7
				7	PRSP_REG_U_MAN	Régulateur de tension en mode Manuel	7
				8	PRSP_REG_U_COS	Régulateur de tension Cosj En	7
				9	PRSP_EXC_CMD_LOC	Excitation en mode Local	7
				10	PRSP_EXC_CMD_DIST	Excitation en mode Distance	7
				11			
				12			
				13			
				14			
15							
%MW 207	400207	4		0	S16_VERR_GR_EPE	Verrouillage de Groupe par EPE Barre	7
				1	PRSP_YM_EN	YM en service	7
				2	PRSP_YR_EN	YR en service	7
				3			
				4			
				5			
				6			
				7			
				8			
				9			
				10			
				11			
				12			
				13	S1_COMMUT_A_M	Magelis : Commutation auto - manuel	*7
				14	S1_COMMUT_M_A	Magelis : Commutation manuel - auto	*7
15	S1_COMMUT_IMP	Magelis : Commutation impossible	*7				
%MW 208	400208	5		0	S4_TU_REPOS	Groupe au repos	7
				1	S4_START	Démarrage du Groupe	7
				2	P_Lub_E	Enclenchement de la pompe de lubrification	7
				3			
				4			
				5	P_Dem_E	Enclenchement de la pompe de démarrage	7
				6			
				7			
				8	P_Ref_E	Enclenchement de la pompe de réfrigération	7
				9			
				10			
				11	SCD_E_LM_H	Ouverture du déflecteur et limiteur mécanique	7
				12			
				13			
				14	SCP_H	Ouverture du pointeau	7
15							
%MW 209	400209	5		0			
				1	Exc_EN	Enclenchement de l'excitation	7
				2			
				3			
				4	Gr_ON	Groupe en service	7
				5			
				6			
				7			
				8			
				9			
				10			
				11			
				12			
				13			
				14			
15	START	Séquence de démarrage active	7				
%MW 210	400210	6		0	S5_AN	Arrêt Normal du Groupe	7
				1	Red_P	Réduction de la puissance	7
				2			
				3			
				4	Disj_H	Ouverture du disjoncteur 13kV	7
				5			
				6			
				7	Re_U_H	Déclenchement du régulateur de tension	7
				8			
				9			
				10	SCP_SCD_H	Désactivation de l'électrovanne du pointeau et du déflecteur	7
				11			
				12			
				13	P_DEM_H	Déclenchement de la pompe de démarrage	7
				14			
15							
%MW 211	400211	6		0	AN_OUT	Déclenchement de la réfrigération	7
				1	S6_VI_TU_0	Vitesse nulle atteinte	7
				2	MGP_H	Déclenchement de la lubrification	7
				3			
				4			
				5			
				6			
				7			
				8			
				9			
				10			
				11			
				12			
				13			
				14	S13_P_MIN_STOP	P minimale atteinte	7
15	AN	Séquence d'Arrêt Normal active	7				

ERT

EPE

EAG

EAG

EAG

EAG

Miéville Gr2 : Liste des signaux pour Superviseur (sans Horodatage)

Adresse	ADD SCA	MSTR	ADD MSTR	Bit	Libelle	Designation	Classe
%MW 212	400212	7		0	S5_AR	Arrêt Rapide du Groupe	7
				1	AR_IN	Fermeture de la partie hydraulique	7
				2			
				3			
				4	Disj_H_a	Ouverture du disjoncteur 13kV	7
				5			
				6			
				7	Re_U_H_a	Déclenchement du régulateur de tension	7
				8			
				9			
				10	P_DEM_H	Déclenchement de la pompe de démarrage	7
				11			
				12			
				13	AR_OUT	Déclenchement de la réfrigération	7
				14			
15	AR	Séquence d'Arrêt Rapide active	7				
%MW 213	400213	7		0	S5_AU	Arrêt d'Urgence du Groupe	7
				1	AU_IN	Fermeture hydraulique et ouverture Disjoncteur 13kV	7
				2			
				3			
				4	P_DEM_H	Déclenchement de la pompe de démarrage	7
				5			
				6			
				7	AU_OUT	Déclenchement de la réfrigération	7
				8			
				9			
				10			
				11			
				12			
				13			
				14			
15	AU	Séquence d'Arrêt d'Urgence active	7				
%MW 214	400214	8		0	S4_CMD_MOD_DIST	Modes de commande en distance	7
				1	S5_ARRET_GR	Pas d'arrêt de groupe actif	7
				2	S4_AC_DC_OK	Distribution AC-DC ok	7
				3	S4_SURV_OK	Surveillance tension ok	7
				4	S4_TP_OK	Disjoncteurs de mesure ok	7
				5			
				6			
				7			
				8	PRSP_OUT1_MAT_A	Matrice A : Sortie 1	7
				9	PRSP_OUT2_MAT_A	Matrice A : Sortie 2	7
				10	PRSP_OUT3_MAT_A	Matrice A : Sortie 3	7
				11	PRSP_OUT4_MAT_A	Matrice A : Sortie 4	7
				12	PRSP_OUT5_MAT_A	Matrice A : Sortie 5	7
				13	PRSP_OUT6_MAT_A	Matrice A : Sortie 6	7
				14	PRSP_OUT7_MAT_A	Matrice A : Sortie 7	7
15	PRSP_OUT8_MAT_A	Matrice A : Sortie 8	7				
%MW 215	400215	8		0	PRSP_CONT_CHAUF	Retour contacteur Chauffage	7
				1	PRSP_CONT_REFRIG	Retour contacteur Réfrigération En	7
				2	PRSP_CONT_LUBRIF	Retour contacteur Lubrification En	7
				3	PRSP_CONT_DYN_1	Retour contacteur CNA Dynapur 1	7
				4	PRSP_CONT_DYN_2	Retour contacteur CNA Dynapur 2	7
				5	PRSP_CONT_PMP_DEM	Retour contacteur pompe de démarrage En	7
				6	PRSP_CONT_PMP_SERV	Retour contacteur pompe de service En	7
				7	PRSP_CONT_CHAUF_PMP_D	Retour contacteur chauffage pompe de démarrage	7
				8	PRSP_CONT_CHAUF_PMP_S	Retour contacteur chauffage pompe de service	7
				9	PRSP_CONT_PMP1	Retour contacteur pompe 1 En	7
				10	PRSP_CONT_PMP2	Retour contacteur pompe 2 En	7
				11			
				12			
				13			
				14			
15							

EAG

EAG

EAG

EAG

* --> seulement sur écran tactile Magelis

Mesure	Adresse	ADD SCA	Libelle	Designation	
0					
1	%MW700	400700	SPMB_TEMP1_ENR_STA	Température 1 enroulement statorique	
2	%MW701	400701	SPMB_TEMP2_ENR_STA	Température 2 enroulement statorique	
3	%MW702	400702	SPMB_TEMP3_ENR_STA	Température 3 enroulement statorique	
4	%MW703	400703	SPMB_TEMP4_ENR_STA	Température 4 enroulement statorique	
5	%MW704	400704	SPMB_TEMP5_ENR_STA	Température 5 enroulement statorique	
6	%MW705	400705	SPMB_TEMP6_ENR_STA	Température 6 enroulement statorique	
7	%MW706	400706	SPMB_TEMP1_AIR_FR	Température 1 flux d'air frais	
8	%MW707	400707	SPMB_TEMP2_AIR_FR	Température 2 flux d'air frais	
9	%MW708	400708	SPMB_TEMP1_AIR_CH	Température 1 flux d'air chaud	
10	%MW709	400709	SPMB_TEMP2_AIR_CH	Température 2 flux d'air chaud	
11	%MW710	400710	SPMB_TEMP1_CIR_MAG	Température 1 circuit magnétique	
12	%MW711	400711	SPMB_TEMP2_CIR_MAG	Température 2 circuit magnétique	
13	%MW712	400712	SPMB_TEMP3_CIR_MAG	Température 3 circuit magnétique	
14	%MW713	400713	SPMB_TEMP_PAL_TU	Température palier turbine	
15	%MW714	400714	SPMB_TEMP_PAL_EXC	Température palier excitation	MPM
16	%MW715	400715	SPMB_TEMP_H_LUB_IN	Temp. lubrification avant réfrigération	
17	%MW716	400716	SPMB_TEMP_H_LUB_OUT	Temp. lubrification après réfrigération	
18	%MW717	400717	SPMB_TEMP_EAU_IN	Température eau réfrigération entrée	
19	%MW718	400718	SPMB_TEMP_EAU_OUT1	Temp. réfrigération sortie Alternateur	
20	%MW719	400719	SPMB_TEMP_EAU_OUT2	Temp. réfrigération sortie Lubrification	
21	%MW720	400720	SPMB_TEMP_H_REG	Température huile régulateur	
22	%MW721	400721	SPMB_VIB_ARB_TU_X	Vibration arbre turbine axe X	
23	%MW722	400722	SPMB_VIB_ARB_TU_Y	Vibration arbre turbine axe Y	
24	%MW723	400723	SPMB_VIB_ARB_EXC_X	Vibration arbre excitatrice axe X	
25	%MW724	400724	SPMB_VIB_ARB_EXC_Y	Vibration arbre excitatrice axe Y	
26	%MW725	400725	SPMB_VIB_PAL_TU_X	Vibration palier turbine axe X	
27	%MW726	400726	SPMB_VIB_PAL_EXC_X	Vibration palier excitatrice axe X	
28	%MW727	400727	SPMB_VIB_PAL_TU_Z	Vibration palier turbine axe Z	
29	%MW728	400728	SPMB_VIB_ARB_Y	Vibration arbre axial Y	
30	%MW800	400800	PRSP_PRES_REFR	Pression Eau Réfrigération	MER
31	%MW801	400801	PRSP_VITESSE_ROT_ARB	Vitesse de rotation de l'arbre	MTU
32	%MW802	400802	PRSP_I_EXC	Courant d'excitation	
33	%MW803	400803	PRSP_U_EXC	Tension d'excitation	ERT
34	%MW804	400804	PRSP_F_PUIS_EXC	Facteur de puissance	
35	%MW810	400810	PRSP_PRES_CF	Pression de la conduite forcée	EAU
36	%MW811	400811	PRSP_PRES_HUILE_REG	Pression huile régulateur	
37	%MW812	400812	PRSP_PRES_POINTEAU	Pression variable pointeau	
38	%MW813	400813	PRSP_PRES_DEFL	Pression variable déflecteur	
39	%MW814	400814	PRSP_LIMIT_MEC	Position limiteur mécanique	
40	%MW815	400815	PRSP_VITESSE_GR	Vitesse Groupe	
41	%MW816	400816	PRSP_CONS_CH	Consigne de charge	MTU
42	%MW817	400817	PRSP_POS_LIMITEUR	Position limiteur électrique	
43	%MW818	400818	PRSP_POS_POINTEAU	Position pointeau	
44			PRSP_P_ACTIVE	Puissance active	
45	%MW820	400820	SPPR_YR	Consigne YR	
46	%MW821	400821	SPPR_YM	Consigne YM	
47	%MW830	400830	PRSP_PRES_1	Pression 1	MVS
48	%MW831	400831	PRSP_PRES_2	Pression 2	
49	%MW850	400850	PRSP_U1_N	Tension U1N	
50	%MW851	400851	PRSP_U2_N	Tension U2N	
51	%MW852	400852	PRSP_U3_N	Tension U3N	
52	%MW853	400853	PRSP_U12	Tension U12	
53	%MW854	400854	PRSP_U23	Tension U23	
54	%MW855	400855	PRSP_U31	Tension U31	EPE
55	%MW856	400856	PRSP_I1	Courant I1	
56	%MW857	400857	PRSP_I2	Courant I2	
57	%MW858	400858	PRSP_I3	Courant I3	
58	%MW859	400859	PRSP_P	Puissance active	
59	%MW860	400860	PRSP_Q	Puissance réactive	
60	%MW861	400861	PRSP_F	Fréquence	

Compteurs

1	%MW900	400900		Heures de fonctionnement de la pompe de démarrage
2	%MW904	400904		Nombre d'enclenchements de la pompe de démarrage
3	%MW908	400908		Heures de fonctionnement de la pompe de service
4	%MW912	400912		Nombre d'enclenchements de la pompe de service
5	%MW916	400916		Heures de fonctionnement de la pompe 1 MVS
6	%MW920	400920		Nombre d'enclenchements de la pompe 1 MVS
7	%MW924	400924		Heures de fonctionnement de la pompe 2 MVS
8	%MW928	400928		Nombre d'enclenchements de la pompe 2 MVS
9	%MW932	400932		Heures de fonctionnement de la pompe de lubrification
10	%MW936	400936		Nombre d'enclenchements de la pompe de lubrification
11	%MW940	400940		Heures de fonctionnement de la pompe de réfrigération
12	%MW944	400944		Nombre d'enclenchements de la pompe de réfrigération
13	%MW948	400948		Heures de service du disjoncteur de Groupe
14	%MW952	400952		Nombre d'enclenchements du disjoncteur de Groupe

Mesure	Adresse	ADD SCA	Libelle	Designation
Paramètres				
1	%MW101	400101	MBSP PAR 1 MPM	Alarme Température enroulements statoriques 1 à 6
2	%MW102	400102	MBSP PAR 2 MPM	Alarme Température flux d'air frais 1 et 2
3	%MW103	400103	MBSP PAR 3 MPM	Alarme Température flux d'air chaud 1 et 2
4	%MW104	400104	MBSP PAR 4 MPM	Alarme Température circuits magnétiques 1 à 3
5	%MW105	400105	MBSP PAR 5 MPM	Alarme Température palier turbine
6	%MW106	400106	MBSP PAR 6 MPM	Alarme Température palier excitation
7	%MW107	400107	MBSP PAR 7 MPM	Alarme Température huile de régulation
8	%MW108	400108	MBSP PAR 8 MPM	Alarme Température lubrification avant réfrigération
9	%MW109	400109	MBSP PAR 9 MPM	Alarme Température lubrification après réfrigération
10	%MW110	400110	MBSP PAR 10 MPM	Alarme Température eau réfrigération entrée
11	%MW111	400111	MBSP PAR 11 MPM	Alarme Température réfrigération sortie Alternateur
12	%MW112	400112	MBSP PAR 12 MPM	Alarme Température réfrigération sortie Lubrification
13	%MW113	400113	MBSP PAR 13 MPM	
14	%MW114	400114	MBSP PAR 14 MPM	
15	%MW115	400115	MBSP PAR 15 MPM	
16	%MW116	400116	MBSP PAR 16 MPM	Déclenchement Température enroulements statoriques 1 à 6
17	%MW117	400117	MBSP PAR 17 MPM	Déclenchement Température flux d'air frais 1 et 2
18	%MW118	400118	MBSP PAR 18 MPM	Déclenchement Température flux d'air chaud 1 et 2
19	%MW119	400119	MBSP PAR 19 MPM	Déclenchement Température circuits magnétiques 1 à 3
20	%MW120	400120	MBSP PAR 20 MPM	Déclenchement Température palier turbine
21	%MW121	400121	MBSP PAR 21 MPM	Déclenchement Température palier excitation
22	%MW122	400122	MBSP PAR 22 MPM	Déclenchement Température huile de régulation
23	%MW123	400123	MBSP PAR 23 MPM	Déclenchement Température lubrification avant réfrigération
24	%MW124	400124	MBSP PAR 24 MPM	Déclenchement Température lubrification après réfrigération
25	%MW125	400125	MBSP PAR 25 MPM	Déclenchement Température eau réfrigération entrée
26	%MW126	400126	MBSP PAR 26 MPM	Déclenchement Température réfrigération sortie Alternateur
27	%MW127	400127	MBSP PAR 27 MPM	Déclenchement Température réfrigération sortie Lubrification
28	%MW128	400128	MBSP PAR 28 MPM	
29	%MW129	400129	MBSP PAR 29 MPM	
30	%MW130	400130	MBSP PAR 30 MPM	
31	%MW131	400131	MBSP PAR 31 MPM	Temps de surveillance pour circulation d'huile de lubrification
32	%MW132	400132	MBSP PAR 32 MPM	Temps de surveillance pour circulation d'eau de réfrigération
33	%MW133	400133	MBSP PAR 33 MPM	Temps de surveillance pour pression d'huile de régulation
34	%MW134	400134	MBSP PAR 34 MPM	Temps de surveillance pour pression d'huile de l'agrégat vanne
35	%MW135	400135	MBSP PAR 35 MPM	
36	%MW136	400136	MBSP PAR 36 MPM	
37	%MW137	400137	MBSP PAR 37 MPM	
38	%MW138	400138	MBSP PAR 38 MPM	
39	%MW139	400139	MBSP PAR 39 MPM	
40	%MW140	400140	MBSP PAR 40 MPM	
41	%MW141	400141	MBSP PAR 41 MPM	
42	%MW142	400142	MBSP PAR 42 MPM	
43	%MW143	400143	MBSP PAR 43 MPM	
44	%MW144	400144	MBSP PAR 44 MPM	
45	%MW145	400145	MBSP PAR 45 MPM	
46	%MW146.0	400146.0	MBSP PAR 46 MPM	Désactivation de la pompe de service de régulation
47	%MW146.1	400146.1	MBSP PAR 47 MPM	Synchro Dead Bus autorisée
48	%MW146.2	400146.2	MBSP PAR 48 MPM	Séquentiel en mode pas à pas
49	%MW146.3	400146.3	MBSP PAR 49 MPM	
50	%MW146.4	400146.4	MBSP PAR 50 MPM	
51	%MW146.5	400146.5	MBSP PAR 51 MPM	
52	%MW146.6	400146.6	MBSP PAR 52 MPM	
53	%MW146.7	400146.7	MBSP PAR 53 MPM	
54	%MW146.8	400146.8	MBSP PAR 54 MPM	
55	%MW146.9	400146.9	MBSP PAR 55 MPM	
56	%MW146.10	400146.10	MBSP PAR 56 MPM	
57	%MW146.11	400146.11	MBSP PAR 57 MPM	
58	%MW146.12	400146.12	MBSP PAR 58 MPM	
59	%MW146.13	400146.13	MBSP PAR 59 MPM	
60	%MW146.14	400146.14	MBSP PAR 60 MPM	Activation Test survitesse

Anhang 5

framework.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE HYDRO_P [
  <!ATTLIST item
    id ID #REQUIRED
  >
]>
<HYDRO_P xmlns="http://www.hydro.ch" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:x="http://www.hydro.ch" xsi:schemaLocation="http://www.hydro.ch framework.xsd" >
  <!-- SCADA part of the project -->
  <scada>
    <group name="MTU" >
      <node name="TURB" >
        <m>
          </m>
        <s>
          <point description="Rotationsgeschwindigkeit" datatype="INT" x:semantic="speed" ref="ID-000001" />
          <point description="ROT bei 80 Prozent" alarm_class="AL2P" datatype="BOOL" x:semantic="speed_ind" ref="ID-000003" />
          <point description="ROT bei 95 Prozent" alarm_class="AL2P" datatype="BOOL" x:semantic="speed_ind" ref="ID-000002" />
        </s>
        <a>
          <point description="Notabschaltung (120 Prozent)" alarm_class="AL1P" alarm_msg="Notabschaltung"
            datatype="BOOL" x:semantic="button_ind" ref="ID-000004" />
        </a>
      </node>
    <node name="DEFIL" >
      <p>
        <point description="Ablenker max Position" datatype="INT" x:semantic="distance" ref="ID-000051" />
        <point description="Ablenker min Position" datatype="INT" x:semantic="distance" ref="ID-000052" />
      </p>
      <s>
        <point description="Ablenker max Position erreicht" alarm_class="SP2L" x:semantic="distance_ind" datatype="BOOL" ref="ID-000007" />
        <point description="Ablenker min Position erreicht" alarm_class="SP2L" x:semantic="distance_ind" datatype="BOOL" ref="ID-000006" />
        <point description="Ablenker ein/aus" alarm_class="SP3L" x:semantic="on/off" datatype="BOOL" ref="ID-000008" />
      </s>
    </node>
  <node name="LIMI" >
```

```
<rs>
  <point description="Limitator el. max Pos erreicht" alarm_class="AL2p" x:semantic="position max" datatype="BOOL" ref="ID-000015" />
  <point description="Limitator el. min Pos erreicht" alarm_class="AL2p" x:semantic="position min" datatype="BOOL" ref="ID-000016" />
  <point description="Limitator mech. geschlossen" alarm_class="AL2p" x:semantic="open_ind" datatype="BOOL" ref="ID-000017" />
  <point description="Limitator mech. offen" alarm_class="AL2p" x:semantic="close_ind" datatype="BOOL" ref="ID-000018" />
</rs>
<a>
  </a>
</node>
</group>
<group name="BAR">
  <node name="BARSALANFE">
    <m>
      <point description="Wasserhfüllstand" datatype="INT" x:semantic="level" ref="ID-000015" />
    </m>
  </node>
</group>
</scada>
<!-- connected fields on Project -->
<field>
  <agent id="1" name="Siemens Field" driver="Siemens TCP/IP Ethernet" port="Ethernet">
    <device id="11" name="DEVSiemens1" devicetype="S7-300">
      <address>
        <modbus/>
        <ip address="153.109.5.240" />
      </address>
      <items>
        <collection name="Rack1">
          <item id="ID-000002" name="PRSP_MIP_VIT_95" datatype="Boolean" rights="Read Only" >
            <description>slot 4</description>
          </item>
        </collection>
      </items>
    </device>
  </agent>
</field>
</group>
</scada>
```

```
<address>I0.1</address>
</item>
<item id="ID-000003" name="PRSP_MIP_VIT_80" datatype="Boolean" rights="Read Only">
  <description>slot 5</description>
</item>
<item id="ID-000004" name="PRSP_MIP_VIT_120" datatype="Boolean" rights="Read Only">
  <description>slot 6</description>
</item>
</collection>
<collection name="Rack2">
  <item id="ID-000006" name="PRSP_SCD_FIRM" datatype="Boolean" rights="Read Only">
    <description>slot 6</description>
  </item>
  <item id="ID-000007" name="PRSP_SCD_OUV" datatype="Boolean" rights="Read Only">
    <description>slot 6</description>
  </item>
  <item id="ID-000008" name="TCSP_SCD_CMD" datatype="Boolean" rights="Read Only">
    <description>slot 6</description>
  </item>
  <item id="ID-000051" name="TCSP_SCD_POSMAX" datatype="Word" rights="Read/Write">
    <description>slot 6</description>
  </item>
  <item id="ID-000052" name="TCSP_SCD_POSMIN" datatype="Word" rights="Read/Write">
    <description>slot 6</description>
  </item>
</collection>
<item id="ID-000005" name="PRSP_PRES_DEFL" datatype="Word" rights="Read Only">
  <description>Rack 3 slot 6</description>
</item>
```

```
</item>
<item id="ID-000001" name="PRSP_VIT_ROT_ARB" datatype="Word" rights="Read Only">
  <description>Rack 3 slot 3 </description>
  <address>IW272</address>
</item>

</items>
</device>
</agent>

<agent id="2" name="Modbus Field" driver="Modbus Ethernet" port="Ethernet" portnb="11">

<device id="21" name="DEVQuantum1" devicetype="Modbus Eth Device" >
<address>
<modbus address="1" />
<ip address="153.109.5.241" />
</address>
</items>
<item id="ID-000012" name="PRSP_CONT_EDN_L_M_E" datatype="Boolean" x:semantic="contact" rights="Read Only">
  <description>Rack 1 Slot 5 </description>
  <address>I1.5.3 => I3</address>
</item>
<item id="ID-000013" name="PRSP_CONT_EDN_L_M_F" datatype="Boolean" x:semantic="contact" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.4 => I00004 </address>
</item>
<item id="ID-000014" name="PRSP_CONT_EDN_L_M_O" datatype="Boolean" x:semantic="over voltage" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.5 => I5</address>
</item>
<item id="ID-000015" name="PRSP_LIM_EL_MAX" datatype="Boolean" x:semantic="position max" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.2 => I00002 </address>
</item>
<item id="ID-000016" name="PRSP_LIM_EL_MIN" datatype="Boolean" x:semantic="position min" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.6 => I6</address>
</item>
<item id="ID-000017" name="PRSP_LIM_MEC_FERM" datatype="Boolean" x:semantic="close_ind" rights="Read Only">
  <description>Rack 1 slot 5 </description>
```

```
<address>I1.5.7 => I7</address>
</item>
<item id="ID-000018" name="PRSP_LIM_MEC_OUV" datatype="Boolean" x:semantic="open_ind" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.8 => I8</address>
</item>
<item id="ID-000019" name="PRSP_TH_LIM_MEC" datatype="Boolean" x:semantic="temperature" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.9 => I9</address>
</item>
<item id="ID-000020" name="TCSP_CYL_MOINS" datatype="Boolean" x:semantic="command" rights="Read Only">
  <description>Rack 1 slot 5 </description>
  <address>I1.5.12 => I12</address>
</item>
<item id="ID-000021" name="TCSP_CYL_PLUS" datatype="Boolean" x:semantic="command" rights="Read/Write">
  <description>Rack 1 slot 2 </description>
  <address>I1.5.13 => I13</address>
</item>
<item id="ID-000022" name="TCSP_LIM_MEC_CMD_F" datatype="Boolean" x:semantic="close_ind" rights="Read/Write">
  <description>Rack 4 slot 2 </description>
  <address>I1.5.14 => I14</address>
</item>
<item id="ID-000023" name="TCSP_LIM_MEC_CMD_O" datatype="Boolean" x:semantic="open_ind" rights="Read/Write">
  <description>Rack 3 slot 1 </description>
  <address>I1.3.15 => I15</address>
</item>
<item id="ID-000025" name="BAR_NIV_EAU" datatype="Word" x:semantic="level" rights="Read Only">
  <description>Rack 3 slot 1 </description>
  <address>MW325</address>
</item>
</items>
</device>
</agent>
</field>
</HYDRO_P>
```

Anhang 6

XSLTCimplicity.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">
  <xsl:output method="text" indent="no" media-type="text" />

  <xsl:template match="/">
    <xsl:variable name="Cimplicity"
select="concat('Cimplicity','/','Points','/','import_points','.CSV')" />
    <xsl:value-of select="$Cimplicity" /> <!-- Creating-->
    <xsl:result-document href="{ $Cimplicity}" >

<xsl:text>PT_ID,ACCESS,ACCESS_FILTER,ACK_TIMEOUT,ADDR,ADDR_OFFSET,ADDR_TYPE,ALM_CLASS,ALM_CRIT
ERIA,</xsl:text>

<xsl:text>ALM_DEADBAND,ALM_DELAY,ALM_ENABLE,ALM_HIGH_1,ALM_HIGH_2,ALM_HLP_FILE,ALM_LOW_1,ALM_L
OW_2,</xsl:text>

<xsl:text>ALM_MSG,ALM_ROUTE_OPER,ALM_ROUTE_SYSMGR,ALM_ROUTE_USER,ALM_SEVERITY,ALM_STR,ALM_TYPE
,</xsl:text>

<xsl:text>ALM_UPDATE_VALUE,ANALOG_DEADBAND,BFR_COUNT,BFR_DUR,BFR_EVENT_PERIOD,BFR_EVENT_PT_ID,
BFR_EVENT_TYPE,</xsl:text>

<xsl:text>BFR_EVENT_UNITS,BFR_GATE_COND,BFR_SYNC_TIME,CALC_TYPE,CONV_LIM_HIGH,CONV_LIM_LOW,CON
V_TYPE,DELAY_LOAD,DELETE_REQ,</xsl:text>

<xsl:text>DESC,DEVIATION_PT,DEVICE_ID,DISP_LIM_HIGH,DISP_LIM_LOW,DISP_TYPE,DISP_WIDTH,ELEMENTS
,ENG_UNITS,ENUM_ID,EQUATION,EXTRA,</xsl:text>

<xsl:text>FW_CONV_EQ,GR_SCREEN,INIT_VAL,JUSTIFICATION,LEVEL,LOCAL,LOG_ACK,LOG_DATA,LOG_DEL,LOG
_GEN,LOG_RESET,MAX_STACKED,MEASUREMENT_UNIT_ID,</xsl:text>

<xsl:text>MISC_FLAGS,POLL_AFTER_SET,PRECISION,PROC_ID,PTMGMT_PROC_ID,PT_ENABLED,PT_ORIGIN,PT_S
ET_INTERVAL,PT_SET_TIME,PT_TYPE,RANGE_HIGH,RANGE_LOW,</xsl:text>

<xsl:text>RAW_LIM_HIGH,RAW_LIM_LOW,REP_TIMEOUT,RESET_ALLOWED,RESET_COND,RESET_PT,RESET_TIMEOUT
,RESOURCE_ID,REV_CONV_EQ,ROLLOVER_VAL,SAFETY_PT,SAMPLE_INTV,</xsl:text>

<xsl:text>SAMPLE_INTV_UNIT,SCAN_RATE,SETPOINT_HIGH,SETPOINT_LOW,TIME_OF_DAY,TRIG_CHK_PT,TRIG_PT
,TRIG_REL,TRIG_VAL,UAFSET,UPDATE_CRITERIA,VARIANCE_VAL,VAR</xsl:text>
      <xsl:text>&#13;&##&#13;&#10;</xsl:text>
      <xsl:apply-templates select="//fr:point" />

    </xsl:result-document>
  </xsl:template>

  <xsl:template match="//fr:point">
    <!-- PT_ID-->
    <xsl:value-of select="ancestor::fr:group/@name" /><xsl:text>.</xsl:text>
    <xsl:value-of select="ancestor::fr:node/@name" /><xsl:text>.</xsl:text>
    <xsl:value-of select="name(..)" /><xsl:text>.</xsl:text>
    <xsl:value-of select="id(@ref)/@name" />
    <xsl:text>,</xsl:text>
    <!-- ACCESS-->
    <xsl:if test="id(@ref)/@rights='Read Only'"><xsl:text>R</xsl:text></xsl:if>
    <xsl:if test="id(@ref)/@rights='Read/Write'"><xsl:text>W</xsl:text></xsl:if>
    <xsl:text>,</xsl:text>
    <!-- ACCESS_FILTER-->
    <xsl:text>,</xsl:text>
    <!-- ACK_TIMEOUT-->

```

```

<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>-1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ADDR -->
<xsl:text>${</xsl:text><xsl:value-of
select="ancestor::fr:group/@name" /><xsl:text>]</xsl:text>
<xsl:value-of select="id(@ref)/ancestor::fr:agent/@name" /><xsl:text>.</xsl:text>
<xsl:value-of select="id(@ref)/ancestor::fr:device/@name" /><xsl:text>.</xsl:text>
<xsl:if test="id(@ref)/ancestor::fr:collection/@name!='' "><xsl:value-of
select="id(@ref)/ancestor::fr:collection/@name" /><xsl:text>.</xsl:text></xsl:if>
<xsl:if test="id(@ref)/ancestor::fr:collection/@name='' "></xsl:if>
<xsl:value-of select="id(@ref)/@name" />
<xsl:text>,</xsl:text>
<!-- ADDR_OFFSET -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- ADDR_TYPE -->
<xsl:text>FQ</xsl:text>
<xsl:text>,</xsl:text>
<!-- ALM_CLASS -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:value-of select="@alarm_class" /><!-- ALARM CLASS TO BE DEFINED!!!! -->
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_CRITERIA -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>ABS</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_DEADBAND -->
<xsl:text>,</xsl:text>
<!-- ALM_DELAY -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- ALM_ENABLE -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_HIGH_1 -->
<xsl:text>,</xsl:text>
<!-- ALM_HIGH_2 -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_HLP_FILE -->
<xsl:text>,</xsl:text>
<!-- ALM_LOW_1 -->
<xsl:text>,</xsl:text>
<!-- ALM_LOW_2 -->
<xsl:text>,</xsl:text>
<!-- ALM_MSG -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <!-- if alarm_msg is empty take the description as alarm message otherwise take
the alarm_msg -->
  <xsl:if test="string-length(@alarm_msg) = 0">
    <xsl:value-of
select="ancestor::fr:group/@name" /><xsl:text>:</xsl:text><xsl:value-of select="@description" />
  </xsl:if>
  <xsl:if test="string-length(@alarm_msg) != 0">
    <xsl:value-of
select="ancestor::fr:group/@name" /><xsl:text>:</xsl:text><xsl:value-of select="@alarm_msg" />

```

```
</xsl:if>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_ROUTE_OPER -->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_ROUTE_SYSMGR -->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_ROUTE_USER -->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_SEVERITY -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- ALM_STR -->
<xsl:text>1</xsl:text>
<xsl:text>,</xsl:text>
<!-- ALM_TYPE -->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>AL</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- ALM_UPDATE_VALUE -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- ANALOG_DEADBAND -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_COUNT -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_DUR -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_EVENT_PERIOD -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_EVENT_PT_ID -->
<xsl:text>,</xsl:text>
<!-- BFR_EVENT_TYPE -->
<xsl:text>1</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_EVENT_UNITS -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_GATE_COND -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- BFR_SYNC_TIME -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- CALC_TYPE -->
<xsl:text>,</xsl:text>
<!-- CONV_LIM_HIGH -->
<xsl:text>,</xsl:text>
<!-- CONV_LIM_LOW -->
<xsl:text>,</xsl:text>
```

```

<!-- CONV_TYPE -->
<xsl:text>NO</xsl:text>
<xsl:text>,</xsl:text>
<!-- DELAY_LOAD -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- DELETE_REQ -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>AR</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- DESC -->
<xsl:value-of select="@description"/>
<xsl:text>,</xsl:text>
<!-- DEVIATION_PT -->
<xsl:text>,</xsl:text>
<!-- DEVICE_ID -->
<xsl:text></xsl:text> <!-- has to be setz with the -clie import import.csv -D <name
given on the device (OPC SERVER) in cimplicity> -->
<xsl:text>,</xsl:text>
<!-- DISP_LIM_HIGH -->
<xsl:text>,</xsl:text>
<!-- DISP_LIM_LOW -->
<xsl:text>,</xsl:text>
<!-- DISP_TYPE -->
<xsl:text>d</xsl:text>
<xsl:text>,</xsl:text>
<!-- DISP_WIDTH -->
<xsl:text>,</xsl:text>
<!-- ELEMENTS -->
<xsl:text>1</xsl:text>
<xsl:text>,</xsl:text>
<!-- ENG_UNITS -->
<xsl:text>,</xsl:text>
<!-- ENUM_ID -->
<xsl:text>,</xsl:text>
<!-- EQUATION -->
<xsl:text>,</xsl:text>
<!-- EXTRA -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- FW_CONV_EQ -->
<xsl:text>,</xsl:text>
<!-- GR_SCREEN -->
<xsl:text>,</xsl:text>
<!-- INIT_VAL -->
<xsl:text>,</xsl:text>
<!-- JUSTIFICATION -->
<xsl:text>LEFT</xsl:text>
<xsl:text>,</xsl:text>
<!-- LEVEL -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- LOCAL -->
<xsl:text>,</xsl:text>
<!-- LOG_ACK -->
<xsl:if test="name(..)='a' or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- LOG_DATA -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- LOG_DEL -->

```

```

<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>0</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- LOG_GEN-->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- LOG_RESET-->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- MAX_STACKED-->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>0</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- MEASUREMENT_UNIT_ID-->
<xsl:text>,</xsl:text>
<!-- MISC_FLAGS-->
<xsl:text>00</xsl:text>
<xsl:text>,</xsl:text>
<!-- POLL_AFTER_SET-->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- PRECISION-->
<xsl:text>,</xsl:text>
<!-- PROC_ID-->
<xsl:text>,</xsl:text>
<!-- PTMGMT_PROC_ID-->
<xsl:text>,</xsl:text>
<!-- PT_ENABLED-->
<xsl:text>1</xsl:text>
<xsl:text>,</xsl:text>
<!-- PT_ORIGIN-->
<xsl:text>D</xsl:text>
<xsl:text>,</xsl:text>
<!-- PT_SET_INTERVAL-->
<xsl:text>,</xsl:text>
<!-- PT_SET_TIME-->
<xsl:text>,</xsl:text>
<!-- PT_TYPE-->
<xsl:value-of select="@datatype"/> <!-- DATATYPES SUPPORTED from KEPWARE to Cimplicity
-->
<xsl:text>,</xsl:text>
<!-- RANGE_HIGH-->
<xsl:text>,</xsl:text>
<!-- RANGE_LOW-->
<xsl:text>,</xsl:text>
<!-- RAW_LIM_HIGH-->
<xsl:text>,</xsl:text>
<!-- RAW_LIM_LOW-->
<xsl:text>,</xsl:text>
<!-- REP_TIMEOUT-->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>1</xsl:text>
</xsl:if>
<xsl:text>,</xsl:text>
<!-- RESET_ALLOWED-->
<xsl:if test="name(..)='a'or name(..)='s' ">
  <xsl:text>0</xsl:text>
</xsl:if>

```

```
<xsl:text>,</xsl:text>
<!-- RESET_COND -->
<xsl:text>,</xsl:text>
<!-- RESET_PT -->
<xsl:text>,</xsl:text>
<!-- RESET_TIMEOUT -->
<xsl:text>,</xsl:text>
<!-- RESOURCE_ID -->
<xsl:value-of select="ancestor::fr:group/@name"/><!-- RESOURCE ID equal the
logicalDevice name in the Framework.xml -->
<xsl:text>,</xsl:text>
<!-- REV_CONV_EQ -->
<xsl:text>,</xsl:text>
<!-- ROLLOVER_VAL -->
<xsl:text>,</xsl:text>
<!-- SAFETY_PT -->
<xsl:text>,</xsl:text>
<!-- SAMPLE_INTV -->
<xsl:text>0</xsl:text>
<xsl:text>,</xsl:text>
<!-- SAMPLE_INTV_UNIT -->
<xsl:text>,</xsl:text>
<!-- SCAN_RATE -->
<xsl:text>1</xsl:text>
<xsl:text>,</xsl:text>
<!-- SETPOINT_HIGH -->
<xsl:text>,</xsl:text>
<!-- SETPOINT_LOW -->
<xsl:text>,</xsl:text>
<!-- TIME_OF_DAY -->
<xsl:text>,</xsl:text>
<!-- TRIG_CHK_PT -->
<xsl:text>,</xsl:text>
<!-- TRIG_PT -->
<xsl:text>,</xsl:text>
<!-- TRIG_REL -->
<xsl:text>NO</xsl:text>
<xsl:text>,</xsl:text>
<!-- TRIG_VAL -->
<xsl:text>,</xsl:text>
<!-- UAFSET -->
<xsl:text>,</xsl:text>
<!-- UPDATE_CRITERIA -->
<xsl:text>UC</xsl:text> <!-- Unsolicited On Changed for -->
<xsl:text>,</xsl:text>
<!-- VARIANCE_VAL -->
<xsl:text>,</xsl:text>
<!-- VARS -->
<xsl:text>1</xsl:text>
<xsl:text>&#13;&#10;</xsl:text>

</xsl:template>

</xsl:stylesheet>
```

Anhang 7

servermain.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified" xmlns="http://www.kepware.com/schemas/servermain"
  xmlns:servermain="http://www.kepware.com/schemas/servermain"
  targetNamespace="http://www.kepware.com/schemas/servermain">

  <!--*****Simple Types*****-->

  <xs:simpleType name="name_t">
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
      <!-- Names can be up to 256 chars but cannot begin with quote, underscore or spaces and
cannot -->
      <!-- contain any dots -->
      <xs:pattern value="^[^_&quot;\s][^.] {0,255}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="comport_t">
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
      <!-- COM 1-999 or None (None is accepted for compability with legacy XML -->
      <!-- Newer format uses nocoms_t defined below) -->
      <xs:pattern value="None|COM\s+[1-9][0-9]{0,2}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="virtualnetwork_t">
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
      <!--Network 1-50 or None -->
      <xs:pattern value="None|Network\s+([1-4][0-9]|50|[1-9])"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="transactioncount_t">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="99"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="baud_t">
    <xs:restriction base="xs:unsignedInt">
      <xs:enumeration value="300"/>
      <xs:enumeration value="600"/>
      <xs:enumeration value="1200"/>
      <xs:enumeration value="2400"/>
      <xs:enumeration value="4800"/>
      <xs:enumeration value="9600"/>
      <xs:enumeration value="14400"/>
      <xs:enumeration value="19200"/>
      <xs:enumeration value="28800"/>
      <xs:enumeration value="38400"/>
      <xs:enumeration value="56000"/>
      <xs:enumeration value="57600"/>
      <xs:enumeration value="115200"/>
      <xs:enumeration value="128000"/>
      <xs:enumeration value="256000"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="databits_t">
    <xs:restriction base="xs:unsignedInt">
```

```
<xs:enumeration value="5" />
<xs:enumeration value="6" />
<xs:enumeration value="7" />
<xs:enumeration value="8" />
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="parity_t">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse" />
    <xs:enumeration value="None" />
    <xs:enumeration value="Odd" />
    <xs:enumeration value="Even" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="stopbits_t">
  <xs:restriction base="xs:unsignedInt">
    <xs:enumeration value="1" />
    <xs:enumeration value="2" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="flowcontrol_t">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse" />
    <xs:enumeration value="None" />
    <xs:enumeration value="DTR" />
    <xs:enumeration value="RTS" />
    <xs:enumeration value="RTS, DTR" />
    <xs:enumeration value="RTS Always" />
    <xs:enumeration value="RTS Manual" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="manualrts_t">
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="9999" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="idletimeout_t">
  <xs:restriction base="xs:unsignedInt">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="99" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="modemconfig_t">
  <xs:sequence>
    <xs:element name="PermanentLineID" type="xs:unsignedLong" />
    <xs:element name="DevConfig" type="xs:base64Binary" />
    <xs:element name="ConnectTimeoutSec" default="60" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
          <xs:minInclusive value="0" />
          <xs:maxInclusive value="9999" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="AutoDial" type="xs:boolean" default="false" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

```

<xs:complexType name="nocomms_t">
  <xs:sequence>
    <xs:element name="FailReads" default="0" minOccurs="0" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="serial_t">
  <xs:choice>
    <!-- Use this sequence when defining a direct serial connection (or None) -->
    <xs:sequence>
      <xs:element name="Port" type="comport_t"/>
      <xs:element name="Baud" type="baud_t"/>
      <xs:element name="DataBits" type="databits_t"/>
      <xs:element name="Parity" type="parity_t"/>
      <xs:element name="StopBits" type="stopbits_t"/>
      <xs:element name="FlowControl" type="flowcontrol_t"/>
      <xs:element name="ManualRTSRaiseMS" default="10" minOccurs="0" type="manualrts_t"/>
      <xs:element name="ManualRTSDropMS" default="10" minOccurs="0" type="manualrts_t"/>
      <xs:element name="ManualRTSDelayMS" default="10" minOccurs="0" type="manualrts_t"/>
      <xs:element name="ReportCommErrors" type="xs:boolean" default="true" minOccurs="0"/>
      <!-- UseModem is a legacy element maintained for compatibility only -->
      <xs:element name="UseModem" type="xs:boolean" default="false" minOccurs="0"/>
      <xs:element name="CloseWhenIdle" type="xs:boolean" default="true" minOccurs="0"/>
      <xs:element name="IdleTimeoutSec" type="idletimeout_t" default="15" minOccurs="0"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element name="Modem" type="modemconfig_t"/>
      <xs:element name="ReportCommErrors" type="xs:boolean" default="true" minOccurs="0"/>
      <xs:element name="CloseWhenIdle" type="xs:boolean" default="true" minOccurs="0"/>
      <xs:element name="IdleTimeoutSec" type="idletimeout_t" default="15" minOccurs="0"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element name="NoCommunication" type="nocomms_t"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

<!--*****Hierarchy*****-->
<xs:element name="Project">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Title" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="64"/>
            <xs:whiteSpace value="preserve"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Comments" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="4096"/>
            <xs:whiteSpace value="preserve"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Options" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Section" maxOccurs="unbounded">

```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="Value" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:anySimpleType">
            <xs:attribute name="Name">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:whiteSpace value="preserve" />
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Name">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:whiteSpace value="preserve" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element ref="AliasList" minOccurs="0" />
<xs:element ref="GlobalDriverSettingsList" minOccurs="0" />
<xs:element ref="ChannelList" minOccurs="0" />
<xs:element ref="PlugInList" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="AliasList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Alias" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Alias">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="MappedTo">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ScanRateMilliseconds" default="0" minOccurs="0">
        <xs:simpleType>
```

```
<xs:restriction base="xs:unsignedInt">
  <xs:minInclusive value="0" />
  <xs:maxInclusive value="99999990" />
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GlobalDriverSettingsList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GlobalDriverSettings" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:any namespace="##other" processContents="strict" minOccurs="0"
maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ChannelList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Channel" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Channel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="name_t" />
      <xs:element name="Driver">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:choice minOccurs="0">
        <xs:element name="Ethernet">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="NetworkAdapter" default="Default" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:whiteSpace value="preserve" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Serial" type="serial_t" />
        <xs:element name="EthernetEncapsulation">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="NetworkAdapter" default="Default" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:whiteSpace value="preserve" />
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:simpleType>
</xs:element>
<xs:element name="IP" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Port" default="2101" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="65535" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Protocol" default="TCP/IP" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse" />
      <xs:enumeration value="UDP" />
      <xs:enumeration value="TCP/IP" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ConnectionTimeoutSeconds" default="3" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="1" />
      <xs:maxInclusive value="999" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CloseWhenIdle" type="xs:boolean" default="true" minOccurs="0" />
<xs:element name="IdleTimeoutSec" type="idletimeout_t" default="15"
minOccurs="0" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="UnsolicitedEthernetEncapsulation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NetworkAdapter" default="Default" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Port" default="2101" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="0" />
            <xs:maxInclusive value="65535" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Protocol" default="TCP/IP" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="UDP" />
            <xs:enumeration value="TCP/IP" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
<xs:element name="PhoneNumberList" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PhoneNumber" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:whiteSpace value="preserve" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Number">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:whiteSpace value="collapse" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Description" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:whiteSpace value="preserve" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- ModemConfig at this level is maintained for backwards compability only. -->
<xs:element name="ModemConfig" minOccurs="0" type="modemconfig_t" />
<xs:element name="WriteOptimizations" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Method" default="Write last value only" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Write all values" />
            <xs:enumeration value="Write last value only" />
            <xs:enumeration value="Write last non-boolean value only" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="WritesToRead" default="10" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="1" />
            <xs:maxInclusive value="10" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

</xs:complexType>
</xs:element>
<xs:element name="NonNormalizedFloatHandlingType" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
      <xs:enumeration value="Unmodified"/>
      <xs:enumeration value="Replaced with zero"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CommunicationSerialization" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VirtualNetwork" type="virtualnetwork_t" minOccurs="0"/>
      <xs:element name="TransactionsPerCycle" type="transactioncount_t" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="InterDeviceDelay" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="DelayMS" default="0" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="60000"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Diagnostics" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Enabled" type="xs:boolean" default="false" minOccurs="0"/>
      <xs:element name="Identifier" type="xs:unsignedInt" default="0" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:choice minOccurs="0">
  <xs:element name="CustomChannelProperties" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##other" processContents="strict" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CustomEncodedProperties" type="xs:base64Binary" minOccurs="0"/>
</xs:choice>
  <xs:element ref="DeviceList" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DeviceList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Device" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Device">

```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="Name" type="name_t" />
    <xs:element name="Description" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="0" />
          <xs:maxLength value="255" />
          <xs:whiteSpace value="preserve" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DiagsIdentifier" type="xs:unsignedInt" default="0" minOccurs="0" />
    <xs:element name="Enabled" type="xs:boolean" default="true" minOccurs="0" />
    <xs:element name="Simulated" type="xs:boolean" default="false" minOccurs="0" />
    <xs:element name="ModelInfo" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##other" processContents="strict" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ID" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Format" use="optional" default="String">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:whiteSpace value="preserve" />
                  <xs:enumeration value="Decimal" />
                  <xs:enumeration value="Hex" />
                  <xs:enumeration value="Octal" />
                  <xs:enumeration value="String" />
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="ScanMode" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="UseClientRate" />
          <xs:enumeration value="UseFloorRate" />
          <xs:enumeration value="ForceAllToFloorRate" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="ScanFloorMs" type="xs:unsignedLong" minOccurs="0" />
    <xs:element name="EthernetEncapsulation" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="IP">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:whiteSpace value="collapse" />
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Port" default="2101" minOccurs="0">
            <xs:simpleType>
```

```
<xs:restriction base="xs:unsignedInt">
  <xs:minInclusive value="0" />
  <xs:maxInclusive value="65535" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Protocol" default="TCP/IP" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse" />
      <xs:enumeration value="UDP" />
      <xs:enumeration value="TCP/IP" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Timing" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ConnectionTimeoutSeconds" default="3" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="1" />
            <xs:maxInclusive value="60" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ResponseTimeoutMilliseconds" default="1000" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="50" />
            <xs:maxInclusive value="99999" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FailAfter" default="3" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="1" />
            <xs:maxInclusive value="10" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="InterRequestDelayMilliseconds" default="0" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="0" />
            <xs:maxInclusive value="30000" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AutoDemotion" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Enabled" type="xs:boolean" default="false" minOccurs="0" />
      <xs:element name="DemoteAfter" default="3" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="1" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:maxInclusive value="30" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="DemoteForMilliseconds" default="10000" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="100" />
      <xs:maxInclusive value="3600000" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="DiscardWrites" type="xs:boolean" default="false" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="AutoTagGeneration" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Startup" default="Do not generate on startup" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Do not generate on startup" />
            <xs:enumeration value="Always generate on startup" />
            <xs:enumeration value="Generate on first startup" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="DuplicateTagAction" default="Delete on create" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Delete on create" />
            <xs:enumeration value="Overwrite as necessary" />
            <xs:enumeration value="Do not overwrite" />
            <xs:enumeration value="Do not overwrite, log error" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Group" default="" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="0" />
            <xs:maxLength value="256" />
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="AllowSubGroups" type="xs:boolean" default="true" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AltAddressMapping" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AltAddressMappingEnabled" type="xs:boolean" default="false"
minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TimeSync" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
```

```

<xs:element name="TimeZone" type="xs:string" minOccurs="0" />
<xs:element name="UsingDST" type="xs:boolean" default="false" minOccurs="0" />
<xs:element name="TimeSyncMethod" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedInt">
      <xs:enumeration value="0" />
      <xs:enumeration value="1" />
      <xs:enumeration value="2" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:choice minOccurs="0">
  <xs:element name="SyncIntervalMinutes">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="1" />
        <xs:maxInclusive value="50400" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="SyncAbsolute" type="xs:string" minOccurs="0" />
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:choice minOccurs="0">
  <xs:element name="CustomDeviceProperties" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##other" processContents="strict" minOccurs="0"
maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CustomEncodedProperties" type="xs:base64Binary" minOccurs="0" />
</xs:choice>
<xs:element ref="TagGroupList" minOccurs="0" />
<xs:element ref="TagList" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="TagGroupList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="TagGroup" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TagGroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="name_t" />
      <xs:element name="AutoGenerated" type="xs:boolean" default="false" minOccurs="0" />
      <xs:element ref="TagGroupList" minOccurs="0" />
      <xs:element ref="TagList" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="TagList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Tag" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
</xs:element>
<xs:element name="Tag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="name_t" />
      <xs:element name="AutoGenerated" type="xs:boolean" default="false" minOccurs="0" />
      <xs:element name="Address">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="1" />
            <xs:maxLength value="1024" />
            <xs:whiteSpace value="preserve" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="DataType" default="Default" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Default" />
            <xs:enumeration value="Boolean" />
            <xs:enumeration value="Char" />
            <xs:enumeration value="Byte" />
            <xs:enumeration value="Short" />
            <xs:enumeration value="Word" />
            <xs:enumeration value="Long" />
            <xs:enumeration value="DWord" />
            <xs:enumeration value="Float" />
            <xs:enumeration value="Double" />
            <xs:enumeration value="String" />
            <xs:enumeration value="BCD" />
            <xs:enumeration value="LBCD" />
            <xs:enumeration value="Date" />
            <xs:enumeration value="Boolean Array" />
            <xs:enumeration value="Char Array" />
            <xs:enumeration value="Byte Array" />
            <xs:enumeration value="Short Array" />
            <xs:enumeration value="Word Array" />
            <xs:enumeration value="Long Array" />
            <xs:enumeration value="DWord Array" />
            <xs:enumeration value="Float Array" />
            <xs:enumeration value="Double Array" />
            <xs:enumeration value="String Array" />
            <xs:enumeration value="BCD Array" />
            <xs:enumeration value="LBCD Array" />
            <xs:enumeration value="Date Array" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ReadWriteAccess" default="Read/Write" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Read Only" />
            <xs:enumeration value="Read/Write" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RespectGUIDataType" type="xs:boolean" default="false" minOccurs="0" />
      <xs:element name="ScanRateMilliseconds" default="100" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="10" />
            <xs:maxInclusive value="99999990" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Description" default="" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="0" />
      <xs:maxLength value="255" />
      <xs:whiteSpace value="preserve" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Scaling" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Type" default="None" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Linear" />
            <xs:enumeration value="None" />
            <xs:enumeration value="Square Root" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="RawLow" type="xs:double" default="0" minOccurs="0" />
      <xs:element name="RawHigh" type="xs:double" default="1000" minOccurs="0" />
      <xs:element name="ScaledDataType" default="Double" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="collapse" />
            <xs:enumeration value="Default" />
            <xs:enumeration value="Boolean" />
            <xs:enumeration value="Char" />
            <xs:enumeration value="Byte" />
            <xs:enumeration value="Short" />
            <xs:enumeration value="Word" />
            <xs:enumeration value="Long" />
            <xs:enumeration value="DWord" />
            <xs:enumeration value="Float" />
            <xs:enumeration value="Double" />
            <xs:enumeration value="String" />
            <xs:enumeration value="BCD" />
            <xs:enumeration value="LBCD" />
            <xs:enumeration value="Date" />
            <xs:enumeration value="Boolean Array" />
            <xs:enumeration value="Char Array" />
            <xs:enumeration value="Byte Array" />
            <xs:enumeration value="Short Array" />
            <xs:enumeration value="Word Array" />
            <xs:enumeration value="Long Array" />
            <xs:enumeration value="DWord Array" />
            <xs:enumeration value="Float Array" />
            <xs:enumeration value="Double Array" />
            <xs:enumeration value="String Array" />
            <xs:enumeration value="BCD Array" />
            <xs:enumeration value="LBCD Array" />
            <xs:enumeration value="Date Array" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ScaledLow" type="xs:double" default="0" minOccurs="0" />
      <xs:element name="ScaledHigh" type="xs:double" default="1000" minOccurs="0" />
      <xs:element name="ClampLow" type="xs:boolean" default="false" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="ClampHigh" type="xs:boolean" default="false" minOccurs="0"/>
<xs:element name="Units" default="" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="32"/>
      <xs:whiteSpace value="collapse"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="NegateValue" type="xs:boolean" default="false" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="PlugInList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="PlugIn" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PlugIn">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FriendlyName">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="CustomPlugInProperties" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <!-- *****
                Description:
                Allow the root node of any plug-in if it has a
                different namespace AND the namespace has a schema.
                ***** -->
            <xs:any namespace="##other" processContents="strict" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Anhang 8

XSLTKepware.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
  xmlns:fr="http://www.hydro.ch" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <xsl:output method="html" indent="no"/>

  <xsl:template match="/">
    <xsl:variable name="Kepware" select="concat('Kepware','/', 'project', '.xml')"/>
    <xsl:value-of select="$Kepware"/>
    <!-- Creating -->
    <xsl:result-document href="{ $Kepware }">
      <servermain:Project xmlns:servermain="http://www.kepware.com/schemas/servermain">
        <servermain:Title> </servermain:Title>
        <servermain:Comments> </servermain:Comments>
        <servermain:AliasList> </servermain:AliasList>
        <servermain:GlobalDriverSettingsList>
          <servermain:GlobalDriverSettings xmlns:name="Modbus Ethernet"
xmlns:modbus_ethernet="http://www.kepware.com/schemas/modbus_ethernet">
<modbus_ethernet:GlobalUnsolicitedPort>502</modbus_ethernet:GlobalUnsolicitedPort>

<modbus_ethernet:GlobalUnsolicitedProtocol>TCP/IP</modbus_ethernet:GlobalUnsolicitedProtocol>
          </servermain:GlobalDriverSettings>
        </servermain:GlobalDriverSettingsList>
        <servermain:ChannelList>
          <!--apply template for each driver -->
          <xsl:apply-templates select="fr:HYDRO_P/fr:field"/>
        </servermain:ChannelList>
        <servermain:PlugInList>
          <servermain:PlugIn>
            <servermain:FriendlyName>Connection Sharing
Plug-in</servermain:FriendlyName>
            <servermain:CustomPlugInProperties> </servermain:CustomPlugInProperties>
          </servermain:PlugIn>
        </servermain:PlugInList>
        </servermain:Project>
      </xsl:result-document>
    </xsl:template>

    <!-- apply template for each agent -->
    <xsl:template match="fr:field">
      <xsl:apply-templates select="fr:agent"/>
    </xsl:template>

    <!-- choose automatically the correct driver template -->
    <xsl:template match="fr:agent" xmlns:servermain="http://www.kepware.com/schemas/servermain">

      <servermain:Channel>
        <servermain:Name>
          <xsl:value-of select="@name"/>
        </servermain:Name>
        <servermain:Driver>
          <xsl:value-of select="@driver"/>
        </servermain:Driver>

        <!-- apply template for correct -->
        <xsl:apply-templates select="@port"/>

        <servermain:WriteOptimizations>
          <servermain:Method>Write last value only </servermain:Method>
          <servermain:WritesToRead>10</servermain:WritesToRead>
        </servermain:WriteOptimizations>

```

```

    <servermain:NonNormalizedFloatHandlingType>Replaced with
zero</servermain:NonNormalizedFloatHandlingType>
    <servermain:CommunicationSerialization>
        <servermain:VirtualNetwork>None</servermain:VirtualNetwork>
        <servermain:TransactionsPerCycle>1</servermain:TransactionsPerCycle>
    </servermain:CommunicationSerialization>
    <servermain:InterDeviceDelay>
        <servermain:DelayMS>0</servermain:DelayMS>
    </servermain:InterDeviceDelay>
    <servermain:Diagnostics>
        <servermain:Enabled>>false</servermain:Enabled>
        <servermain:Identifier>
            <xsl:value-of select="@id" />
        </servermain:Identifier>
    </servermain:Diagnostics>
    <!-- apply channel custom template -->
    <xsl:apply-templates select="@driver[.='Modbus Ethernet' ]"/>
    <servermain:DeviceList>

        <!-- apply the template DEVICE on all -->
        <xsl:apply-templates select="fr:device"/>

    </servermain:DeviceList>
</servermain:Channel>
</xsl:template>

<!-- SELECT RIGHT TEMPLATE OF COMMUNICATION (ETHERNET / SERIAL) -->
<xsl:template match="@port">
    <xsl:apply-templates select="@port[.='Ethernet' ]"/>
    <xsl:apply-templates select="@port[.='Serial' ]"/>
</xsl:template>
<!-- ETHERNET TEMPLATE -->
<xsl:template match="@port[.='Ethernet' ]"
xmlns:servermain="http://www.kepware.com/schemas/servermain">
    <servermain:Ethernet>
        <servermain:NetworkAdapter/>
    </servermain:Ethernet>
</xsl:template>
<!-- SERIAL TEMPLATE -->
<xsl:template match="@port[.='Serial' ]"
xmlns:servermain="http://www.kepware.com/schemas/servermain">
    <servermain:Serial>
        <servermain:Port>COM <xsl:value-of select="../@portnb" /></servermain:Port>
        <servermain:Baud>9600</servermain:Baud>
        <servermain:DataBits>8</servermain:DataBits>
        <servermain:Parity>Even</servermain:Parity>
        <servermain:StopBits>1</servermain:StopBits>
        <servermain:FlowControl>None</servermain:FlowControl>
        <servermain:ReportCommErrors>>true</servermain:ReportCommErrors>
        <servermain:CloseWhenIdle>>true</servermain:CloseWhenIdle>
        <servermain:IdleTimeoutSec>15</servermain:IdleTimeoutSec>
    </servermain:Serial>
</xsl:template>

<!-- TEMPLATE FOR CHANNEL CUSTOM PROPRIETES -->
<xsl:template match="@driver[.='Modbus Ethernet' ]"
xmlns:servermain="http://www.kepware.com/schemas/servermain">
    <servermain:CustomChannelProperties
xmlns:modbus_ethernet="http://www.kepware.com/schemas/modbus_ethernet">
        <modbus_ethernet:UseMultipleSockets>>false</modbus_ethernet:UseMultipleSockets>
        <modbus_ethernet:MaxSocketsPerDevice>1</modbus_ethernet:MaxSocketsPerDevice>
    </servermain:CustomChannelProperties>
</xsl:template>

```

```

<!--DEVICE TEMPLATE -->
<xsl:template match="fr:device" xmlns:servermain="http://www.kepware.com/schemas/servermain">

  <servermain:Device>

    <servermain:Name>
      <xsl:value-of select="@name" />
    </servermain:Name>
    <servermain:DiagsIdentifier>
      <xsl:value-of select="@id" />
    </servermain:DiagsIdentifier>
    <servermain:Enabled>true</servermain:Enabled>
    <servermain:Simulated>>false</servermain:Simulated>

    <!-- APPLY TEMPLATES FOR MODEL -->
    <xsl:apply-templates select="@devicetype" />

    <servermain:TagGroupList>

      <xsl:apply-templates select="fr:items/fr:collection" />

    </servermain:TagGroupList>

    <servermain:TagList>

      <xsl:apply-templates select="fr:items/fr:item" />

    </servermain:TagList>

  </servermain:Device>

</xsl:template>

<!-- TEMPLATE FOR EACH SPESIFIC DEVICE -->
<xsl:template match="@devicetype">

  <xsl:apply-templates select="@devicetype[.='Modbus RTU Device' ]" />
  <xsl:apply-templates select="@devicetype[.='Modbus Eth Device' ]" />
  <xsl:apply-templates select="@devicetype[.='S7-200' ]" />
  <xsl:apply-templates select="@devicetype[.='S7-300' ]" />
  <xsl:apply-templates select="@devicetype[.='S7-400' ]" />
  <xsl:apply-templates select="@devicetype[.='S7-1200' ]" />
  <xsl:apply-templates select="@devicetype[.='IEC61850' ]" />

</xsl:template>

<!-- TEMPLATE MODBUS SERIEL CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='Modbus RTU Device' ]"
xmlns:servermain="http://www.kepware.com/schemas/servermain">

  <servermain:ModelInfo xmlns:modbus="http://www.kepware.com/schemas/modbus">
    <modbus:Model>Modbus</modbus:Model>
  </servermain:ModelInfo>

  <servermain:ID Format="Decimal">
    <xsl:value-of select=" ../fr:address/fr:modbus/@address" />
  </servermain:ID>

  <servermain:ScanMode>UseClientRate</servermain:ScanMode>
  <servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

  <servermain:EthernetEncapsulation>
    <servermain:IP>255.255.255.255</servermain:IP>
    <servermain:Port>2101</servermain:Port>
    <servermain:Protocol>TCP/IP</servermain:Protocol>

```

```

</servermain:EthernetEncapsulation>

<servermain:Timing>
  <servermain:ResponseTimeoutMilliseconds>1000</servermain:ResponseTimeoutMilliseconds>
  <servermain:FailAfter>3</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

<servermain:AutoDemotion>
  <servermain:Enabled>false</servermain:Enabled>
  <servermain:DemoteAfter>3</servermain:DemoteAfter>
  <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
  <servermain:DiscardWrites>false</servermain:DiscardWrites>
</servermain:AutoDemotion>

<servermain:AutoTagGeneration>
  <servermain:Startup>Do not generate on startup </servermain:Startup>
  <servermain:DuplicateTagAction>Delete on create </servermain:DuplicateTagAction>
  <servermain:Group/>
  <servermain:AllowSubGroups>true</servermain:AllowSubGroups>
</servermain:AutoTagGeneration>

<servermain:CustomDeviceProperties xmlns:modbus ="http://www.kepware.com/schemas/modbus">
  <modbus:Settings>
    <modbus:ZeroBasedAddressing>true</modbus:ZeroBasedAddressing>

<modbus:ZeroBasedBitAddressingRegisters>true</modbus:ZeroBasedBitAddressingRegisters>
  <modbus:HoldingRegisterBitMaskWrites>true</modbus:HoldingRegisterBitMaskWrites>

<modbus:ModbusFunction06SingleRegisterWrites>true</modbus:ModbusFunction06SingleRegisterWrites>

<modbus:ModbusFunction05SingleCoilWrites>true</modbus:ModbusFunction05SingleCoilWrites>
  <modbus:DefaultModbusByteOrder>true</modbus:DefaultModbusByteOrder>
  <modbus:FirstWordLow32BitDataTypes>true</modbus:FirstWordLow32BitDataTypes>
  <modbus:FirstDWordLow64BitDataTypes>true</modbus:FirstDWordLow64BitDataTypes>
  <modbus:ModiconBitOrdering>false</modbus:ModiconBitOrdering>
  </modbus:Settings>
  <modbus:BlockSizes>
    <modbus:Output>32</modbus:Output>
    <modbus:Input>32</modbus:Input>
    <modbus:Internal>32</modbus:Internal>
    <modbus:Holding>32</modbus:Holding>
    <modbus:PerformBlockStringRead>false</modbus:PerformBlockStringRead>
  </modbus:BlockSizes>
  <modbus:VariableImportSettings>
    <modbus:VariableImportFile>*.txt</modbus:VariableImportFile>
    <modbus:DisplayDescriptions>true</modbus:DisplayDescriptions>
  </modbus:VariableImportSettings>
  <modbus:Framing>
    <modbus:UseTcpFraming>false</modbus:UseTcpFraming>
    <modbus:NumLeadingBytes>0</modbus:NumLeadingBytes>
    <modbus:NumTrailingBytes>0</modbus:NumTrailingBytes>
  </modbus:Framing>
  <modbus:ErrorHandling>

<modbus:DeactivateOnIllegalAddressException>true</modbus:DeactivateOnIllegalAddressException>
  <modbus:RepeatedMessageRejection>false</modbus:RepeatedMessageRejection>
  </modbus:ErrorHandling>
</servermain:CustomDeviceProperties>
</xsl:template>
<!-- TEMPLATE MODBUS ETHERNET CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='Modbus Eth Device']"
xmlns:servermain="http://www.kepware.com/schemas/servermain">

```

```
<servermain:ModelInfo
xmlns:modbus_ethernet="http://www.kepware.com/schemas/modbus_ethernet">
  <modbus_ethernet:Model>Modbus</modbus_ethernet:Model>
</servermain:ModelInfo>

<servermain:ID Format="String">&lt;<xsl:value-of
select="../fr:address/fr:ip/@address"/>&gt;><xsl:value-of
select="../fr:address/fr:modbus/@address"/></servermain:ID>

<servermain:ScanMode>UseClientRate</servermain:ScanMode>
<servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

<servermain:Timing>
  <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
  <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
  <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

<servermain:AutoDemotion>
  <servermain:Enabled>false</servermain:Enabled>
  <servermain:DemoteAfter>3</servermain:DemoteAfter>
  <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
  <servermain:DiscardWrites>false</servermain:DiscardWrites>
</servermain:AutoDemotion>

<servermain:AutoTagGeneration>
  <servermain:Startup>Do not generate on startup </servermain:Startup>
  <servermain:DuplicateTagAction>Delete on create </servermain:DuplicateTagAction>
  <servermain:Group/>
  <servermain:AllowSubGroups>true</servermain:AllowSubGroups>
</servermain:AutoTagGeneration>

<servermain:CustomDeviceProperties
xmlns:modbus_ethernet="http://www.kepware.com/schemas/modbus_ethernet">
  <modbus_ethernet:Communications>
    <modbus_ethernet:Port>502</modbus_ethernet:Port>
    <modbus_ethernet:Protocol>TCP/IP</modbus_ethernet:Protocol>

<modbus_ethernet:CloseTCPSocketOnTimeout>true</modbus_ethernet:CloseTCPSocketOnTimeout>
  </modbus_ethernet:Communications>
  <modbus_ethernet:Sub-Model>Generic Modbus</modbus_ethernet:Sub-Model>
  <modbus_ethernet:Settings>
    <modbus_ethernet:ZeroBasedAddressing>true</modbus_ethernet:ZeroBasedAddressing>

<modbus_ethernet:ZeroBasedBitAddressingRegisters>true</modbus_ethernet:ZeroBasedBitAddressingR
egisters>

<modbus_ethernet:HoldingRegisterBitMaskWrites>true</modbus_ethernet:HoldingRegisterBitMaskWrites>

<modbus_ethernet:ModbusFunction06SingleRegisterWrites>true</modbus_ethernet:ModbusFunction06Si
ngleRegisterWrites>

<modbus_ethernet:ModbusFunction05SingleCoilWrites>true</modbus_ethernet:ModbusFunction05Single
CoilWrites>

<modbus_ethernet:DefaultModbusByteOrder>true</modbus_ethernet:DefaultModbusByteOrder>

<modbus_ethernet:FirstWordLow32BitDataTypes>true</modbus_ethernet:FirstWordLow32BitDataTypes>

<modbus_ethernet:FirstDWordLow64BitDataTypes>true</modbus_ethernet:FirstDWordLow64BitDataTypes>
  <modbus_ethernet:ModiconBitOrdering>false</modbus_ethernet:ModiconBitOrdering>
```

```

</modbus_ethernet:Settings>
<modbus_ethernet:BlockSizes>
  <modbus_ethernet:Output>32</modbus_ethernet:Output>
  <modbus_ethernet:Input>32</modbus_ethernet:Input>
  <modbus_ethernet:Internal>32</modbus_ethernet:Internal>
  <modbus_ethernet:Holding>32</modbus_ethernet:Holding>
</modbus_ethernet:BlockSizes>
<modbus_ethernet:PerformBlockStringRead>false</modbus_ethernet:PerformBlockStringRead>
</modbus_ethernet:BlockSizes>
<modbus_ethernet:VariableImportSettings>
  <modbus_ethernet:VariableImportFile>*.txt</modbus_ethernet:VariableImportFile>
  <modbus_ethernet:DisplayDescriptions>>true</modbus_ethernet:DisplayDescriptions>
</modbus_ethernet:VariableImportSettings>
<modbus_ethernet:DataBadUntilWrite>false</modbus_ethernet:DataBadUntilWrite>
<modbus_ethernet:Timeout>0</modbus_ethernet:Timeout>
<modbus_ethernet:ErrorHandling>

<modbus_ethernet:DeactivateOnIllegalAddressException>true</modbus_ethernet:DeactivateOnIllegal
AddressException>
  </modbus_ethernet:ErrorHandling>
</servermain:CustomDeviceProperties>
</xsl:template>
<!-- TEMPLATE SIEMENS S7-1200 CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='S7-1200']">
xmlns:servermain="http://www.kepware.com/schemas/servermain">

  <servermain:ModelInfo
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
    <siemens_tcpip_ethernet:Model>S7-1200</siemens_tcpip_ethernet:Model>
  </servermain:ModelInfo>

  <servermain:ID Format="String">
    <xsl:value-of select="..//fr:address/fr:ip/attribute::address"/>
  </servermain:ID>

  <servermain:ScanMode>UseClientRate</servermain:ScanMode>
  <servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

  <servermain:Timing>
    <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
    <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
    <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

  <servermain:AutoDemotion>
    <servermain:Enabled>false</servermain:Enabled>
    <servermain:DemoteAfter>3</servermain:DemoteAfter>
    <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
    <servermain:DiscardWrites>false</servermain:DiscardWrites>
  </servermain:AutoDemotion>

  <servermain:CustomDeviceProperties
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
    <siemens_tcpip_ethernet:S71200>
      <siemens_tcpip_ethernet:Communications>
        <siemens_tcpip_ethernet:Port>102</siemens_tcpip_ethernet:Port>
        <siemens_tcpip_ethernet:LinkType>PC</siemens_tcpip_ethernet:LinkType>
        <siemens_tcpip_ethernet:Rack>0</siemens_tcpip_ethernet:Rack>
        <siemens_tcpip_ethernet:Slot>1</siemens_tcpip_ethernet:Slot>
      </siemens_tcpip_ethernet:Communications>
      <siemens_tcpip_ethernet:AddressingOptions>

```

```
<siemens_tcpip_ethernet:UseLittleEndianByteOrder>>false</siemens_tcpip_ethernet:UseLittleEndian
ByteOrder>
    </siemens_tcpip_ethernet:AddressingOptions>
</siemens_tcpip_ethernet:S71200>
</servermain:CustomDeviceProperties>

</xsl:template>
<!-- TEMPLATE SIEMENS S7-400 CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='S7-400']">
xmlns:servermain="http://www.kepware.com/schemas/servermain">

    <servermain:ModelInfo
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
        <siemens_tcpip_ethernet:Model>S7-400</siemens_tcpip_ethernet:Model>
    </servermain:ModelInfo>

    <servermain:ID Format="String">
        <xsl:value-of select="..//fr:address/fr:ip/attribute::address"/>
    </servermain:ID>

    <servermain:ScanMode>UseClientRate</servermain:ScanMode>
    <servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

    <servermain:Timing>
        <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
        <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
        <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

    <servermain:AutoDemotion>
        <servermain:Enabled>>false</servermain:Enabled>
        <servermain:DemoteAfter>3</servermain:DemoteAfter>
        <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
        <servermain:DiscardWrites>>false</servermain:DiscardWrites>
    </servermain:AutoDemotion>

    <servermain:CustomDeviceProperties
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
        <siemens_tcpip_ethernet:S7400>
            <siemens_tcpip_ethernet:Communications>
                <siemens_tcpip_ethernet:Port>102</siemens_tcpip_ethernet:Port>
                <siemens_tcpip_ethernet:LinkType>PC</siemens_tcpip_ethernet:LinkType>
                <siemens_tcpip_ethernet:Rack>0</siemens_tcpip_ethernet:Rack>
                <siemens_tcpip_ethernet:Slot>2</siemens_tcpip_ethernet:Slot>
            </siemens_tcpip_ethernet:Communications>
            <siemens_tcpip_ethernet:AddressingOptions>

<siemens_tcpip_ethernet:UseLittleEndianByteOrder>>false</siemens_tcpip_ethernet:UseLittleEndian
ByteOrder>
            </siemens_tcpip_ethernet:AddressingOptions>
        </siemens_tcpip_ethernet:S7400>
    </servermain:CustomDeviceProperties>

</xsl:template>
<!-- TEMPLATE SIEMENS S7-300 CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='S7-300']">
xmlns:servermain="http://www.kepware.com/schemas/servermain">

    <servermain:ModelInfo
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
        <siemens_tcpip_ethernet:Model>S7-300</siemens_tcpip_ethernet:Model>
    </servermain:ModelInfo>
```

```

<servermain:ID Format="String">
  <xsl:value-of select=" ../fr:address/fr:ip/attribute::address" />
</servermain:ID>

<servermain:ScanMode>UseClientRate</servermain:ScanMode>
<servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

<servermain:Timing>
  <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
  <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
  <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

<servermain:AutoDemotion>
  <servermain:Enabled>>false</servermain:Enabled>
  <servermain:DemoteAfter>3</servermain:DemoteAfter>
  <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
  <servermain:DiscardWrites>>false</servermain:DiscardWrites>
</servermain:AutoDemotion>

<servermain:CustomDeviceProperties
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
  <siemens_tcpip_ethernet:S7300>
    <siemens_tcpip_ethernet:Communications>
      <siemens_tcpip_ethernet:Port>102</siemens_tcpip_ethernet:Port>
      <siemens_tcpip_ethernet:LinkType>PC</siemens_tcpip_ethernet:LinkType>
      <siemens_tcpip_ethernet:Rack>0</siemens_tcpip_ethernet:Rack>
      <siemens_tcpip_ethernet:Slot>2</siemens_tcpip_ethernet:Slot>
    </siemens_tcpip_ethernet:Communications>
    <siemens_tcpip_ethernet:AddressingOptions>

<siemens_tcpip_ethernet:UseLittleEndianByteOrder>>false</siemens_tcpip_ethernet:UseLittleEndian
ByteOrder>
      </siemens_tcpip_ethernet:AddressingOptions>
    </siemens_tcpip_ethernet:S7300>
  </servermain:CustomDeviceProperties>
</xsl:template>
<!-- TEMPLATE SIEMENS S7-200 CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='S7-200']">
xmlns:servermain="http://www.kepware.com/schemas/servermain">

  <servermain:ModelInfo
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
    <siemens_tcpip_ethernet:Model>S7-200</siemens_tcpip_ethernet:Model>
  </servermain:ModelInfo>

  <servermain:ID Format="String">
    <xsl:value-of select=" ../fr:address/fr:ip/attribute::address" />
  </servermain:ID>

  <servermain:ScanMode>UseClientRate</servermain:ScanMode>
  <servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

  <servermain:Timing>
    <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
    <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
    <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
</servermain:Timing>

```

```
<servermain:AutoDemotion>
  <servermain:Enabled>false</servermain:Enabled>
  <servermain:DemoteAfter>3</servermain:DemoteAfter>
  <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
  <servermain:DiscardWrites>false</servermain:DiscardWrites>
</servermain:AutoDemotion>

<servermain:CustomDeviceProperties
xmlns:siemens_tcpip_ethernet="http://www.kepware.com/schemas/siemens_tcpip_ethernet">
  <siemens_tcpip_ethernet:S7200>
    <siemens_tcpip_ethernet:Communications>
      <siemens_tcpip_ethernet:Port>102</siemens_tcpip_ethernet:Port>
      <siemens_tcpip_ethernet:LocalTSAP>19799</siemens_tcpip_ethernet:LocalTSAP>
      <siemens_tcpip_ethernet:RemoteTSAP>19799</siemens_tcpip_ethernet:RemoteTSAP>
    </siemens_tcpip_ethernet:Communications>
    <siemens_tcpip_ethernet:AddressingOptions>

<siemens_tcpip_ethernet:UseLittleEndianByteOrder>false</siemens_tcpip_ethernet:UseLittleEndian
ByteOrder>
    </siemens_tcpip_ethernet:AddressingOptions>
  </siemens_tcpip_ethernet:S7200>
</servermain:CustomDeviceProperties>

</xsl:template>
<!-- TEMPLATE IEC61850 CUSTOM DEVICE PROPRIETES -->
<xsl:template match="@devicetype[.='IEC61850']">
xmlns:servermain="http://www.kepware.com/schemas/servermain">

  <servermain:ModelInfo
xmlns:iec_61850_mms_client="http://www.kepware.com/schemas/iec_61850_mms_client">
    <iec_61850_mms_client:Model>MMS</iec_61850_mms_client:Model>
  </servermain:ModelInfo>
  <servermain:ScanMode>UseClientRate</servermain:ScanMode>
  <servermain:ScanFloorMs>1000</servermain:ScanFloorMs>

  <servermain:Timing>
    <servermain:ConnectionTimeoutSeconds>3</servermain:ConnectionTimeoutSeconds>
    <servermain:ResponseTimeoutMilliseconds>2000</servermain:ResponseTimeoutMilliseconds>
    <servermain:FailAfter>2</servermain:FailAfter>

<servermain:InterRequestDelayMilliseconds>0</servermain:InterRequestDelayMilliseconds>
  </servermain:Timing>

  <servermain:AutoDemotion>
    <servermain:Enabled>false</servermain:Enabled>
    <servermain:DemoteAfter>3</servermain:DemoteAfter>
    <servermain:DemoteForMilliseconds>10000</servermain:DemoteForMilliseconds>
    <servermain:DiscardWrites>false</servermain:DiscardWrites>
  </servermain:AutoDemotion>

  <servermain:CustomDeviceProperties
xmlns:iec_61850_mms_client="http://www.kepware.com/schemas/iec_61850_mms_client">
    <iec_61850_mms_client:Settings>
      <iec_61850_mms_client:AutoConfiguration>
        <iec_61850_mms_client:Source>Device</iec_61850_mms_client:Source>
        <iec_61850_mms_client:SCL>
          <iec_61850_mms_client:SCLFile/>
          <iec_61850_mms_client:SubNetwork/>
          <iec_61850_mms_client:IED/>
          <iec_61850_mms_client:AccessPoint/>
        </iec_61850_mms_client:SCL>
      </iec_61850_mms_client:AutoConfiguration>
      <iec_61850_mms_client:Connection>
        <iec_61850_mms_client:Authentication>
```

```
<iec_61850_mms_client:Enabled>>false</iec_61850_mms_client:Enabled>
<iec_61850_mms_client>Password/>
<iec_61850_mms_client:Encrypted>>true</iec_61850_mms_client:Encrypted>
</iec_61850_mms_client:Authentication>
<iec_61850_mms_client:ServerParameters>

<iec_61850_mms_client:UseOptionalParameters>>true</iec_61850_mms_client:UseOptionalParameters>
<iec_61850_mms_client:AEInvokeID>10</iec_61850_mms_client:AEInvokeID>
<iec_61850_mms_client:AEQualifier>12</iec_61850_mms_client:AEQualifier>
<iec_61850_mms_client:APIInvokeID>10</iec_61850_mms_client:APIInvokeID>

<iec_61850_mms_client:ApplicationID>1,1,1,999,1</iec_61850_mms_client:ApplicationID>
<iec_61850_mms_client:PresentationSelector>00 00 00
01</iec_61850_mms_client:PresentationSelector>
<iec_61850_mms_client:SessionSelector>00
01</iec_61850_mms_client:SessionSelector>
<iec_61850_mms_client:TransportSelector>00
01</iec_61850_mms_client:TransportSelector>
</iec_61850_mms_client:ServerParameters>
<iec_61850_mms_client:ClientParameters>

<iec_61850_mms_client:UseOptionalParameters>>true</iec_61850_mms_client:UseOptionalParameters>
<iec_61850_mms_client:AEInvokeID>10</iec_61850_mms_client:AEInvokeID>
<iec_61850_mms_client:AEQualifier>12</iec_61850_mms_client:AEQualifier>
<iec_61850_mms_client:APIInvokeID>10</iec_61850_mms_client:APIInvokeID>

<iec_61850_mms_client:ApplicationID>1,1,1,999,1</iec_61850_mms_client:ApplicationID>
<iec_61850_mms_client:PresentationSelector>00 00 00
01</iec_61850_mms_client:PresentationSelector>
<iec_61850_mms_client:SessionSelector>00
01</iec_61850_mms_client:SessionSelector>
<iec_61850_mms_client:TransportSelector>00
01</iec_61850_mms_client:TransportSelector>
</iec_61850_mms_client:ClientParameters>
</iec_61850_mms_client:Connection>
<iec_61850_mms_client:Communication>
<iec_61850_mms_client:ReportHandling>
<iec_61850_mms_client:Enabled>>true</iec_61850_mms_client:Enabled>
<iec_61850_mms_client:Buffering>
<iec_61850_mms_client:Enabled>>true</iec_61850_mms_client:Enabled>

<iec_61850_mms_client:MaxValuesPerBuffer>100</iec_61850_mms_client:MaxValuesPerBuffer>

<iec_61850_mms_client:PlaybackRate>2000</iec_61850_mms_client:PlaybackRate>
</iec_61850_mms_client:Buffering>
</iec_61850_mms_client:ReportHandling>
<iec_61850_mms_client:Polling>
<iec_61850_mms_client:Interval>5000</iec_61850_mms_client:Interval>
<iec_61850_mms_client:Level>Logical Node</iec_61850_mms_client:Level>

<iec_61850_mms_client:PollReported>false</iec_61850_mms_client:PollReported>
</iec_61850_mms_client:Polling>
</iec_61850_mms_client:Communication>
<iec_61850_mms_client:Control>
<iec_61850_mms_client:orCat>bay-control</iec_61850_mms_client:orCat>
<iec_61850_mms_client:orIdent/>
<iec_61850_mms_client:ctlNum>0</iec_61850_mms_client:ctlNum>
<iec_61850_mms_client:Test>false</iec_61850_mms_client:Test>
<iec_61850_mms_client:Check>00</iec_61850_mms_client:Check>
</iec_61850_mms_client:Control>
<iec_61850_mms_client:TagGenerationSettings>

<iec_61850_mms_client:DisplayDescriptions>true</iec_61850_mms_client:DisplayDescriptions>
```

```

<iec_61850_mms_client:GenerateDataSets>>false</iec_61850_mms_client:GenerateDataSets>
    <iec_61850_mms_client:LogicalDeviceTags>

<iec_61850_mms_client:ConfigureForPolling>>false</iec_61850_mms_client:ConfigureForPolling>
    <iec_61850_mms_client:ST>>true</iec_61850_mms_client:ST>
    <iec_61850_mms_client:MX>>true</iec_61850_mms_client:MX>
    <iec_61850_mms_client:CO>>true</iec_61850_mms_client:CO>
    <iec_61850_mms_client:SP>>true</iec_61850_mms_client:SP>
    <iec_61850_mms_client:SV>>true</iec_61850_mms_client:SV>
    <iec_61850_mms_client:CF>>true</iec_61850_mms_client:CF>
    <iec_61850_mms_client:DC>>true</iec_61850_mms_client:DC>
    <iec_61850_mms_client:SG>>true</iec_61850_mms_client:SG>
    <iec_61850_mms_client:SE>>true</iec_61850_mms_client:SE>
    <iec_61850_mms_client:EX>>true</iec_61850_mms_client:EX>
    <iec_61850_mms_client:BR>>true</iec_61850_mms_client:BR>
    <iec_61850_mms_client:RP>>true</iec_61850_mms_client:RP>
    <iec_61850_mms_client:LG>>true</iec_61850_mms_client:LG>
    <iec_61850_mms_client:GO>>true</iec_61850_mms_client:GO>
    <iec_61850_mms_client:GS>>true</iec_61850_mms_client:GS>
    <iec_61850_mms_client:MS>>true</iec_61850_mms_client:MS>
    <iec_61850_mms_client:US>>true</iec_61850_mms_client:US>
    </iec_61850_mms_client:LogicalDeviceTags>
    </iec_61850_mms_client:TagGenerationSettings>
    </iec_61850_mms_client:Settings>
</servermain:CustomDeviceProperties>
</xsl:template>

<!-- COLLECTION TEMPLATE -->
<xsl:template match="fr:collection">

    <servermain:TagGroup xmlns:servermain="http://www.kepware.com/schemas/servermain">
        <servermain:Name>
            <xsl:value-of select="@name" />
        </servermain:Name>
        <servermain:TagList>
            <xsl:apply-templates select="fr:item" />
        </servermain:TagList>
    </servermain:TagGroup>

</xsl:template>

<!-- ITEM TEMPLATE -->
<xsl:template match="fr:item">

    <servermain:Tag xmlns:servermain="http://www.kepware.com/schemas/servermain">
        <servermain:Name>
            <xsl:value-of select="@name" />
        </servermain:Name>
        <servermain:Address>

            <!-- if device type is a modbus -->
            <xsl:if test="contains(ancestor::fr:device/@devicetype, 'Modbus')">

                <!--
                if there is => in the address tag check if there is a memory address
on the right side of =>
                if true then convert right side into modbusaddress
                else put right side of <address> tag into the address field of tag
                -->
                <xsl:choose>
                    <xsl:when test="contains(fr:address, ' => ')">
                        <xsl:choose>
                            <xsl:when test="contains(substring-after(fr:address, ' =>
'), 'MW')">

```

```

        <xsl:choose>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'MW'))=5">
                <xsl:text>4</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'MW'))=4">
                <xsl:text>40</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'MW'))=3">
                <xsl:text>400</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'MW'))=2">
                <xsl:text>4000</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'MW'))=1">
                <xsl:text>40000</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
            </xsl:when>
        </xsl:choose>
    </xsl:when>
    <xsl:when test="contains(substring-after(fr:address,' =>
'), 'M')">
        <xsl:choose>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'M'))=5">
                <xsl:text>0</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'M'))=4">
                <xsl:text>00</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'M'))=3">
                <xsl:text>000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'M'))=2">
                <xsl:text>0000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
            </xsl:when>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'M'))=1">
                <xsl:text>00000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
            </xsl:when>
        </xsl:choose>
    </xsl:when>
    <xsl:when test="contains(substring-after(fr:address,' =>
'), 'IW')">
        <xsl:choose>
            <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'IW'))=5">
                <xsl:text>3</xsl:text><xsl:value-of

```

```

select="substring-after(fr:address,'IW')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'IW'))=4">
    <xsl:text>30</xsl:text><xsl:value-of
select="substring-after(fr:address,'IW')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'IW'))=3">
    <xsl:text>300</xsl:text><xsl:value-of
select="substring-after(fr:address,'IW')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'IW'))=2">
    <xsl:text>3000</xsl:text><xsl:value-of
select="substring-after(fr:address,'IW')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'IW'))=1">
    <xsl:text>30000</xsl:text><xsl:value-of
select="substring-after(fr:address,'IW')"/>
    </xsl:when>
    </xsl:choose>
</xsl:when>
<xsl:when test="contains(substring-after(fr:address,' =>
'), 'I')">
    <xsl:choose>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'I'))=5">
    <xsl:text>1</xsl:text><xsl:value-of
select="substring-after(substring-after(fr:address,' => '), 'I')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'I'))=4">
    <xsl:text>10</xsl:text><xsl:value-of
select="substring-after(substring-after(fr:address,' => '), 'I')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'I'))=3">
    <xsl:text>100</xsl:text><xsl:value-of
select="substring-after(substring-after(fr:address,' => '), 'I')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'I'))=2">
    <xsl:text>1000</xsl:text><xsl:value-of
select="substring-after(substring-after(fr:address,' => '), 'I')"/>
    </xsl:when>
    <xsl:when
test="string-length(substring-after(substring-after(fr:address,' => '), 'I'))=1">
    <xsl:text>10000</xsl:text><xsl:value-of
select="substring-after(substring-after(fr:address,' => '), 'I')"/>
    </xsl:when>
    </xsl:choose>
</xsl:when>
<xsl:otherwise>
    <!-- take content right of => to put into address field of
tag -->
    <xsl:value-of select="substring-after(fr:address,' => ')"/>
    </xsl:otherwise>
</xsl:choose>

</xsl:when>
<xsl:otherwise>
<xsl:choose>

```

```

        <xsl:when test="contains(fr:address,'MW')">
            <xsl:choose>
                <xsl:when
test="string-length(substring-after(fr:address,'MW'))=5">
                    <xsl:text>4</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'MW'))=4">
                    <xsl:text>40</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'MW'))=3">
                    <xsl:text>400</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'MW'))=2">
                    <xsl:text>4000</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'MW'))=1">
                    <xsl:text>40000</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>
                </xsl:when>
            </xsl:choose>
        </xsl:when>
        <xsl:when test="contains(fr:address,'M')">
            <xsl:choose>
                <xsl:when
test="string-length(substring-after(fr:address,'M'))=5">
                    <xsl:text>0</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'M'))=4">
                    <xsl:text>00</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'M'))=3">
                    <xsl:text>000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'M'))=2">
                    <xsl:text>0000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
                </xsl:when>
                <xsl:when
test="string-length(substring-after(fr:address,'M'))=1">
                    <xsl:text>00000</xsl:text><xsl:value-of
select="substring-after(fr:address,'M')"/>
                </xsl:when>
            </xsl:choose>
        </xsl:when>
        <xsl:when test="contains(fr:address,'IW')">
            <xsl:choose>
                <xsl:when
test="string-length(substring-after(fr:address,'IW'))=5">
                    <xsl:text>3</xsl:text><xsl:value-of
select="substring-after(fr:address,'MW')"/>

```

```

        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'IW'))=4">
            <xsl:text>30</xsl:text><xsl:value-of
select="substring-after(fr:address, 'MW')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'IW'))=3">
            <xsl:text>300</xsl:text><xsl:value-of
select="substring-after(fr:address, 'MW')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'IW'))=2">
            <xsl:text>3000</xsl:text><xsl:value-of
select="substring-after(fr:address, 'MW')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'IW'))=1">
            <xsl:text>30000</xsl:text><xsl:value-of
select="substring-after(fr:address, 'MW')"/>
        </xsl:when>
    </xsl:choose>
</xsl:when>
<xsl:when test="contains(fr:address, 'I')">
    <xsl:choose>
        <xsl:when
test="string-length(substring-after(fr:address, 'I'))=5">
            <xsl:text>1</xsl:text><xsl:value-of
select="substring-after(fr:address, 'M')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'I'))=4">
            <xsl:text>10</xsl:text><xsl:value-of
select="substring-after(fr:address, 'M')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'I'))=3">
            <xsl:text>100</xsl:text><xsl:value-of
select="substring-after(fr:address, 'M')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'I'))=2">
            <xsl:text>1000</xsl:text><xsl:value-of
select="substring-after(fr:address, 'M')"/>
        </xsl:when>
        <xsl:when
test="string-length(substring-after(fr:address, 'I'))=1">
            <xsl:text>10000</xsl:text><xsl:value-of
select="substring-after(fr:address, 'M')"/>
        </xsl:when>
    </xsl:choose>
</xsl:when>
<xsl:otherwise>
    <!-- nothing invalid memory address -->
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:if>
<!-- if device type is a siemens -->
<xsl:if test="contains(ancestor::fr:device/@devicetype, 'S7')">
    <xsl:value-of select="fr:address"/>
</xsl:if>

```

```
</servermain:Address>
<servermain:DataType>
  <xsl:value-of select="@datatype" />
</servermain:DataType>
<servermain:ReadWriteAccess>
  <xsl:value-of select="@rights" />
</servermain:ReadWriteAccess>
<servermain:RespectGUIDataType>true</servermain:RespectGUIDataType>
<servermain:Description>
  <xsl:value-of select="fr:description" />
</servermain:Description>
</servermain:Tag>

</xsl:template>

</xsl:stylesheet>
```

Anhang 9

XSLTUnity.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">

  <xsl:output method="xml" indent="yes"/>

  <xsl:template match="/">
    <xsl:apply-templates select="//fr:device[contains(@devicetype,'Modbus')]" />
  </xsl:template>

  <xsl:template match="//fr:device[contains(@devicetype,'Modbus')]">
    <xsl:variable name="Unity" select="concat(@name,'/','variables','.xsy')" />
    <xsl:value-of select="$Unity" /> <!-- Creating -->
    <xsl:result-document href="{ $Unity }" >

      <xsl:element name="VariablesExchangeFile">

        <xsl:element name="fileHeder">
          <xsl:attribute name="company">Schneider Automation</xsl:attribute>
          <xsl:attribute name="product">Unity Pro XL V7.0</xsl:attribute>
          <xsl:attribute name="dateTime"><xsl:value-of
select="current-dateTime()" /></xsl:attribute>
          <xsl:attribute name="content">variables of device <xsl:value-of
select="@name" /></xsl:attribute>
          <xsl:attribute name="DTDVersion" />
        </xsl:element>
        <xsl:element name="ContentHeader">
          <xsl:attribute name="name"><xsl:value-of select="@name" /></xsl:attribute>
          <xsl:attribute name="version" />
          <xsl:attribute name="dateTime"><xsl:value-of
select="current-dateTime()" /></xsl:attribute>
        </xsl:element>
        <xsl:element name="dataBlock">
          <!-- apply templates for <items> and <collection> -->
          <xsl:apply-templates select="." />
        </xsl:element>
      </xsl:element>

    </xsl:result-document>
  </xsl:template>

  <!-- TEMPLATE ITEM -->
  <xsl:template match="fr:item">

    <xsl:element name="variables">
      <xsl:attribute name="name"><xsl:value-of select="@name" /></xsl:attribute>
      <xsl:attribute name="typeName">
        <!-- change boolean into EBOOL -->
        <xsl:if test="@datatype='Boolean'"><xsl:text>EBOOL</xsl:text></xsl:if>
        <xsl:if test="@datatype!='Boolean'"><xsl:value-of
select="@datatype"></xsl:value-of></xsl:if>
      </xsl:attribute>
      <xsl:attribute name="topologicalAddress">

        <xsl:choose>
          <!-- select left value of " => " in <address> tag if " => " exists -->
          <xsl:when test="contains(fr:address,' => ')">
            <xsl:text>%</xsl:text><xsl:value-of
select="substring-before(fr:address,' => ')" />
          </xsl:when>
          <!-- otherwise select value of <address> tag (should be MWxxx, Mxxx, IWxxx

```

or Ixxx)-->

```
    <xsl:otherwise>
      <xsl:text>%</xsl:text><xsl:value-of select="fr:address" />
    </xsl:otherwise>
  </xsl:choose>

</xsl:attribute>
<xsl:element name="comment"><xsl:value-of select="fr:description" /></xsl:element>
<!-- additional information if it is boolean -->
<xsl:if test="@datatype='Boolean' ">
  <xsl:element name="attribute">
    <xsl:attribute name="name">TimeStampCapability</xsl:attribute>
    <xsl:attribute name="value">0</xsl:attribute>
  </xsl:element>
  <xsl:element name="attribute">
    <xsl:attribute name="name">TimeStampSource</xsl:attribute>
    <xsl:attribute name="value">3</xsl:attribute>
  </xsl:element>
  <xsl:element name="variableInit">
    <xsl:attribute name="value"></xsl:attribute>
  </xsl:element>
</xsl:if>
</xsl:element>

</xsl:template>

</xsl:stylesheet>
```

Anhang 10

XSLTTIAPortal.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">

  <xsl:output method="text" />

  <xsl:template match="/">
    <xsl:apply-templates select="//fr:device[contains(@devicetype,'S7')]" />
  </xsl:template>

  <xsl:template match="//fr:device[contains(@devicetype,'S7')]">
    <xsl:variable name="TIAPortal" select="concat(@name,'/','PLC Tags','.csv')" />
    <xsl:value-of select="$TIAPortal" /> <!-- Creating -->
    <xsl:result-document href="{ $TIAPortal }" >
      <!-- column titel separedet by semicolon -->
      <xsl:text>Name;Path;Data Type;Logical Address;Comment;Hmi Visible;Hmi
Accessible&#13;</xsl:text>
      <!-- apply templates for <items> and <collection> -->
      <xsl:apply-templates select="." />
    </xsl:result-document>
  </xsl:template>

  <!-- TEMPLATE ITEM -->
  <xsl:template match="fr:item">

    <xsl:value-of select="@name" />
    <xsl:text>;</xsl:text>
    <xsl:value-of select="@datatype" />
    <xsl:text>;%</xsl:text>
    <xsl:value-of select="fr:address" />
    <xsl:text>;</xsl:text>
    <xsl:value-of select="fr:description" />
    <xsl:text>;True;True&#13;</xsl:text>

  </xsl:template>

</xsl:stylesheet>
```

Anhang 11

XSLTOPCGroups.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <xsl:variable name="Cimplicity"
select="concat('Cimplicity','/', 'httpdocs', '/', 'port_config', '.html') " />
    <xsl:value-of select="$Cimplicity" /> <!-- Creating-->
    <xsl:result-document href="{ $Cimplicity}" >

      <html>
      <head>
      <h1>OPC Port configuration </h1>
      <style type="text/css">
        body {font-family:calibir; font-size:15px}
        h2 { color:blue; font-family:calibri; font-size: 15px;}
        h1 {font-family:calibri; font-size:20px}
        div {
          width : 200px;
          padding : 0.2em;
          margin : 0.3em;
          border-width :1px;
          border-style :solid;
          border-color :gray;
        }
      </style>
      </head>

      <body>
      <p>
copy the content below and paste it into the <br/>
      <xsl:text><![CDATA[<projectname>]]>/data/<![CDATA[<master_opc_x.ini>]]></xsl:text>
document on the end.
      <br/>
      the x correspond to the selected port number in the cimplicity project under
Equipment / Ports.
      </p>
      <!-- apply template for each logicalDevice in the framework -->
      <xsl:apply-templates select="//fr:group"/>
      </body>
      </html>
    </xsl:result-document>
  </xsl:template>

  <!-- TEMPLATE FOR LOGICAL DEVICE -->
  <xsl:template match="//fr:group">
  <!-- each logicalDevice is considered as an OPCGroup created from Cimplicity (OPC
Client) into Kepware (OPC Server) -->

  <div>
    <h2>[<xsl:value-of select="@name"/>]</h2>
    ScanRate=1000 <br/>
    DeviceReadAfterSet=1 <br/>
    NoAccessPath=0 <br/>
    DelayBeforeRefresh=0 <br/><br/>
  </div>
  </xsl:template>

</xsl:stylesheet>

```

Anhang 12

XSLTAlarmclasses.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">

  <!-- Output file format -->
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <xsl:variable name="Cimplicity"
select="concat('Cimplicity','/', 'httpdocs','/', 'alarm_classes','html')" />
    <xsl:value-of select="$Cimplicity" /> <!-- Creating -->
    <xsl:result-document href="{ $Cimplicity }" >
      <html>
        <title>Alarm classes for Cimplicity </title>
        <head>
          <h1>Alarm classes </h1>

          <style type="text/css">
            body {font-family:calibir; font-size:15px}
            h2 { color:blue; font-family:calibri; font-size: 15px;}
            h1 {font-family:calibri; font-size:20px}
            p { font-family:calibri; font-size:15px}
          </style>
        </head>
        <body>

          <p>the following alarm classes need to be created in the cimplicity
project</p>

          <!-- select all alarmclasses defined in the framework (distinct) -->
          <p>
          <xsl:for-each select="distinct-values(//fr:point/@alarm_class[.!=''])">

            <h2>--> <xsl:value-of select="."/;></h2>

          </xsl:for-each>
          </p>
        </body>
      </html>
    </xsl:result-document>

  </xsl:template>

</xsl:stylesheet>

```

Anhang 13

XSLTResources.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fr="http://www.hydro.ch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0">

  <!-- Output file format -->
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <xsl:variable name="Cimplicity"
select="concat('Cimplicity','/', 'httpdocs','/', 'resources','.html')" />
    <xsl:value-of select="$Cimplicity" /> <!-- Creating -->
    <xsl:result-document href="{ $Cimplicity }" >
      <html>
        <title>Resources for Cimplicity </title>
        <head>
          <h1>Resources</h1>

          <style type="text/css">
            body {font-family:calibir; font-size:15px}
            h2 { color:blue; font-family:calibri; font-size: 15px;}
            h1 {font-family:calibri; font-size:20px}
            p { font-family:calibri; font-size:15px}
          </style>
        </head>
        <body>

          <p>the following resources need to be created in the cimplicity project </p>
          <!-- select all alarmclasses defined in the framework (distinct) -->
          <p>
            <xsl:for-each select="distinct-values(//fr:group/@name[.!=''])">

              <h2>-> <xsl:value-of select="."/></h2>

            </xsl:for-each>
          </p>
        </body>
      </html>
    </xsl:result-document>

  </xsl:template>

</xsl:stylesheet>
```