Real Time Computersoftware Ges.m.b.H

RTAPLS V7.2
Real Time Architecture
Prozeßleitsystem

Kurzbeschreibung

Autor: Christian Riedl, RTS
Stand: 2007/11/02

1 Allgemeines

RTAPLS ist ein Softwareprodukt zur Verarbeitung und Präsentation von Prozessdaten in Prozessleitsystemen. Der Haupteinsatzbereich von RTAPLS sind komplexe, kontinuierliche Prozesse in der Chemie, der Energieerzeugung und -verteilung. RTAPLS ist von der Leistungsfähigkeit hinsichtlich Funktionalität und Datenumfang am oberen Ende derartiger Systeme angesiedelt.

Die wesentlichen Informationseinheiten derartiger Systeme sind Prozessvariable (=PVs). Diese und sämtliche RTAPLS Objekte werden in der Real Time Database (RtDb) speicherresident gehalten. Die RtDb kann auf viele Rechner durch Replikation verteilt werden. Damit sind RTAPLS Objekte wie die PVs verteilte Objekte. Auf den Servern werden diese Objekte dauerhaft (optional in ORACLE DB) gespeichert. Insofern ist das Datenmodell von RTAPLS als relational zu bezeichnen. Grundsätzlich gibt es 3 Typen von PVs. Analoge, binäre (bis zu 16 bit) und textmäßige.

Typische RTAPLS Installationen enthalten 10.000 bis 100.000 Prozessvariable und verarbeiten mehrere tausend Prozessdatenänderungen pro Sekunde.

Die Kernfunktionalitäten von RTAPLS sind hoch performant, plattformunabhängig und objektorientiert in ISO C++ implementiert.

RTAPLS stellt für einfache Applikationsentwicklung auf der Microsoft Plattform Schnittstellen auf der Basis von COM, .NET, und OLEDB zur Verfügung.

Standardisierte Web Services (W3C) erlauben die Integration in die unternehmensweite, serviceorientierte Datenverarbeitung (SOA).

RTAPLS implementiert die OPC (OLE for Process Control) Standards und ermöglicht so die problemlose Integration von I/O Systemen sowie den Einsatz weit verbreiterter Visualisierungs-Werkzeuge.

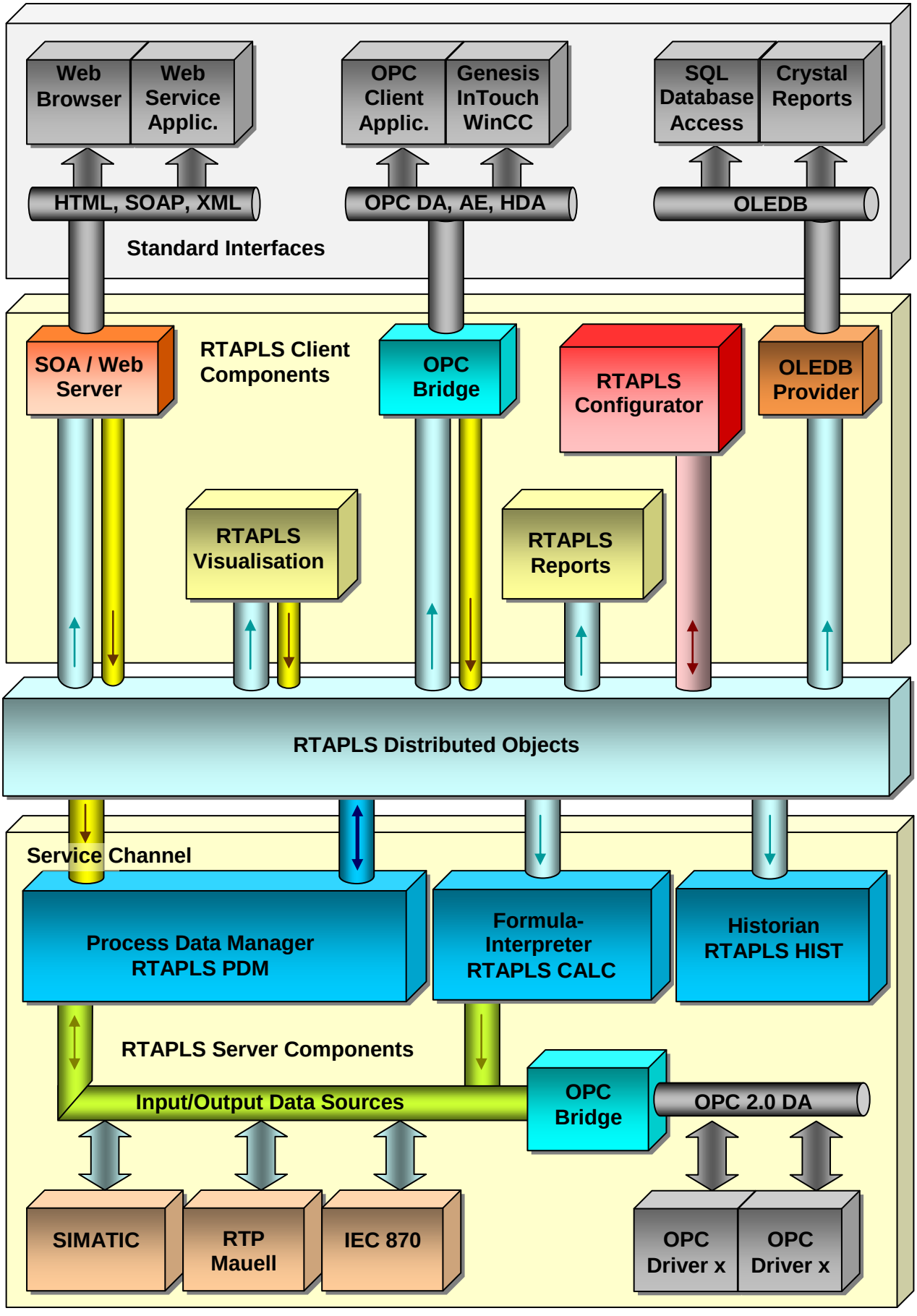
RTAPLS ist multiplattformfähig und auf UNIX / Linux Plattformen, auf VMS und den Microsoft Windows Plattformen verfügbar. Selbst heterogene Konfigurationen (UNIX oder VMS Server und Microsoft Windows Clients) sind möglich.

RTAPLS kann online, ohne Betriebsunterbrechung konfiguriert werden.

Bei Ausfall eines Verarbeitungsrechners werden die Funktionen rückwirkungsfrei vom verbleibenden Rechner weiter bereitgestellt.

RTAPLS stellt umfangreiche Diagnosewerkzeuge zur Verfügung, die Systemzustände, Programmzustände, Fehlersituationen etc. anzeigen und auswerten können.

1.1 RTAPLS Systemarchitektur



2 RTAPLS Komponenten

- Die Echtzeitplattform RTA (Real Time Architecture)
- Das verteilte RTAPLS Objektmodell (RTAPLS - DBO)
- Der Prozessdatenmanager (RTAPLS - PDM)
- Der Formelinterpreter (RTAPLS – CALC)
- Der Historian (RTAPLS – HIST)
- Das Auskunfts- und Berichtssystem (RTAPLS – REPORT)
- Die Prozessvisualisierung (RTAPLS – VIEW)
- Der RTAPLS SOA Server (RTAPLS – SOA)
- Das objektorientierte Engineeringsystem (RtOOS)
- Die RTAPLS Systemadministration (RTAPLS – CONSOLE)
- Die Treiber (Schnittstellen zu den I/O Systemen)

2.1 Integrierte Standards

RTAPLS integriert eine Reihe von marktgängigen Standards :

- ISO C++
- Microsoft Plattform
  - COM/DCOM kompatibles RTA / RTAPLS API
  - .NET kompatibles RTA / RTAPLS API
  - OLE DB Datenbank-Zugriff auf RTAPLS Datenmodell

- OPC (OLE for Process Control) Spezifikationen
  - OPC Data Access DA 1.0, 2.0, 3.0
  - OPC Alarm and Events AE 1.1
  - OPC Historical Data Access HDA 1.2
  - OPC Unified Architecture DA 1.0 (SOA Server)

- ORACLE Datenbank
- Datenimport / Export im XML Format
- Visualisierung basierend auf SVG (Scaleable Vector Graphics)
- W3C Web Service (Service Oriented Architecture)
- SNMP Network Management

2.2 Real Time Architecture – RTA

RTAPLS basiert ganz wesentlich auf der Real Time Architecture (RTA). RTA ist eine Softwareplattform für die Entwicklung von verteilten, redundanten Echtzeitsystemen. Betriebssystemunabhängig und auf Standards basierend werden die für Echtzeitapplikationen relevanten Funktionen zur Verfügung gestellt:

- Real Time Database
- Real Time Interprocess Communication
- Real Time Redundancy Management

RTA unterstützt Unix (HPUX, Linux, Solaris), OpenVMS und Microsoft Windows 2000 / XP / Vista / Server 2003

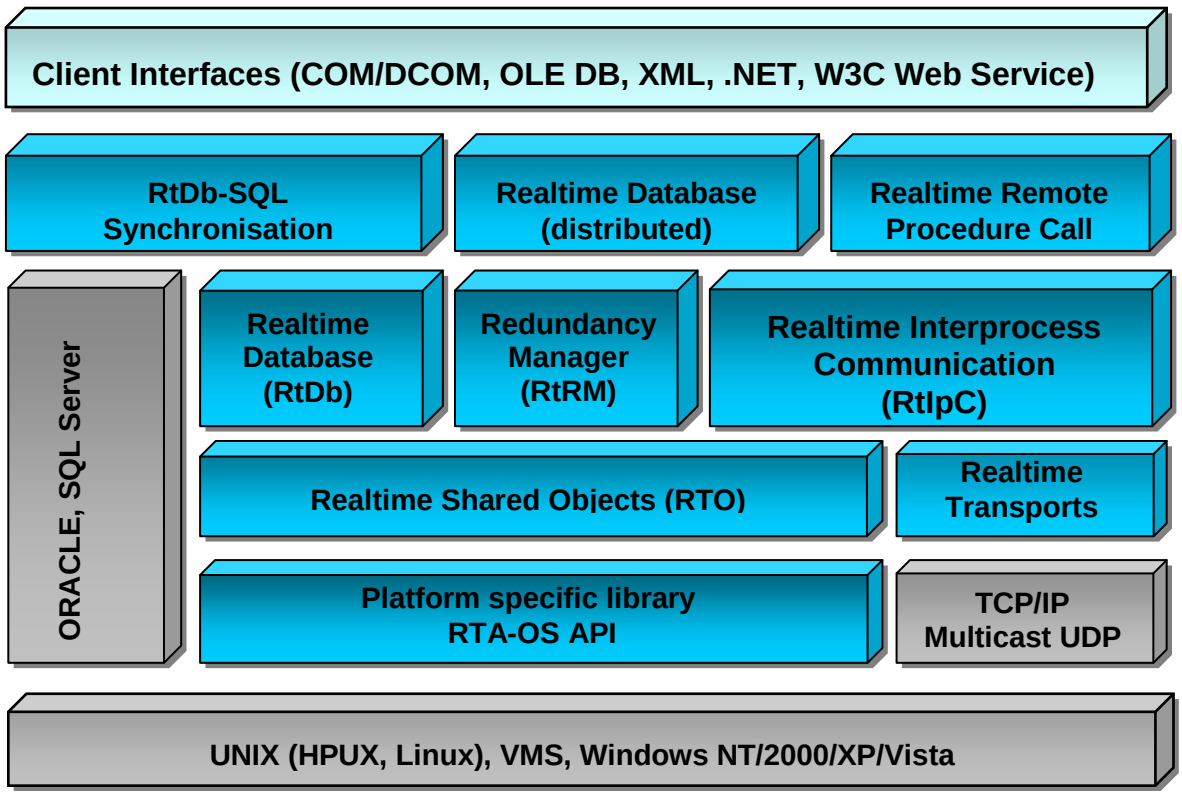
Die Echtzeitfähigkeit von RTA wird in erster Linie durch die Performance von Realtime Database und Interprocess Communication sichergestellt. Reaktionszeiten im Millisekundenbereich sind dadurch ohne weiteres realisierbar.

Die Realtime Database kann synchron zu einer relationalen Datenbank betrieben werden und kann so gleichsam als "Cache" eine relationale Datenbank um etwa den Faktor 100 beschleunigen. Damit kann Datenbanktechnologie mit all ihren Vorteilen auch für Applikationen genutzt werden, die bisher auf konventionelle Mechanismen zurückgreifen mussten.

RTA unterstützt redundante Konfigurationen durch Verdoppelung der Server und unterstützt auch heterogene Konfigurationen. (z.B. VMS-Server oder UNIX-Server, Microsoft Clients).

RTA ist als Softwarewerkzeug in RTAPLS integriert und ermöglicht die Entwicklung von projektspezifschen Applikationen, sie sich nahtlos in die RTAPLS Architektur einfügen.

2.2.1 RTA – Komponenten



2.2.2 Realtime Interprocess Communication (RtIpC)

Die rechnerlokale Kommunikation wird über shared Memory abgewickelt und ermöglicht so lokale Datenraten von über 10.000.000 Prozessdaten-Messages pro Sekunde.

Im Netzwerk können bis zu 50.000 (100 Mb LAN) Prozessdaten-Messages pro Sekunde versendet werden. Eine Besonderheit ist die Unterstützung von redundanten Netzwerken zur Gewährleistung des unterbrechungsfreien Betrieb.

Optimierte Multicast-Protokolle ermöglichen die performante Datenverteilung auf eine große Anzahl von Empfängern.

2.2.3 Realtime Database (RtDb)

Kompromisslose Realtimefähigkeit mit weit über 10.000.000 Einzel-Zugriffen pro Sekunde.

Die Organisation der RtDb ist mit der relationaler Datenbanken vergleichbar. Die Daten werden in Tabellen mit Reihen und Spalten gehalten.

In einem verteilten System kann jeder Node eine lokale Kopie von RtDb-Tabellen führen. Die Synchronisation dieser Tabellen im Änderungsfall erfolgt prioritätsgesteuert mit Hilfe von Multicast Protokollen.

Sogenannte Triggermechanismen erlauben einer Applikation die Definition von Aktionen (Installation von Callback Funktionen), die automatisch aktiviert werden, wenn sich Daten in der RtDb ändern.

Eine automatische Synchronisation der RtDb mit einer relationalen Datenbank (ORACLE oder SQL Server) ist möglich.

2.2.4 Realtime Redundancy Manager (RtRM)

RtRM sorgt für die Überwachung der einzelnen Prozesse und Rechner.

Dieser Prozess steuert auch den Start der Applikationen und im Fehlerfall auch die Rechnerumschaltung bei redundanten Systemen.

Besonders in redundanten Systemen ist die lückenlose Überwachung der lebenswichtigen Softwarefunktionen sehr wichtig, da die Überwachung die Kriterien für eine Rechnerumschaltung liefern muss.

Die einzelnen Prozesse können selber Fehler erkennen und diesen Umstand an RtRM melden. Außerdem besitzt RtRM eine Watchdogfunktion mit der die Prozesse überwacht werden. Das heißt sie müssen sich in bestimmten konfigurierbaren Zyklen melden. Tun sie das nicht, werden sie als fehlerbehaftet betrachtet.

Die einzelnen Rechner überwachen sich gegenseitig durch zyklische Messages auf dem redundanten Netzwerk. Ein Ausfall eines redundanten Rechners führt zu einem Failover. Die Redundanz kann 2 oder mehrere Rechner umfassen. Die Timeoutzeiten sind konfigurierbar.

2.4 Der Prozessdatenmanager (RTAPLS - PDM)

PDM bildet die zentrale Komponente in RTAPLS. Er verarbeitet die Prozessdaten anhand von umfangreichen Konfigurationsdaten. Das Ergebnis dieser Verarbeitungen wird im verteilten RTAPLS Daten- und Objektmodell abgelegt.

Die wichtigsten konfigurierbaren Verarbeitungen sind

- Diverse Skalierungen (linear, kurve, frei programmierbar)
- Überwachung auf 4 Grenzwertpaare und 2 Gradienten
- Konfigurierbare Alarmverarbeitung mit bis zu 8 Alarmprioritäten
  Global und Lokale (pro Workstation) Quittierung
- Automatisch oder manuelle Sperren
- Ersatzwertbildung
- Benutzerspezifisches Berechtigungssystem
- Analoge und binäre Filtermechanismen
  - Glättung
  - Flatterunterdrückung
- Abgeleitete Werte
- Integral, Minimum / Maximum, Mittwelwert Berechnung
- Integriertes, frei programmierbares Mathematikpaket (RTAPLS - CALC)
- Führung von Feldzeit und Erfassungszeit
- Deadband Funktionalität
- 2 fache Anlagen-Hierarchie (mit automatischen Statussummierungen)
- Projektspezifische Statusinformationen
- Umfangreiche Status Information pro PV (eine Untermenge davon OPC kompatibel)
- Rückmeldungsüberwachung

PDM verfügt über 2 Schnittstellen :

- I/O Datenquellen-Schnittstelle für den Transfer von Prozessdaten von und zu den Erfassungssystemen bzw. SPSen. Diese Schnittstelle ist für großen Datendurchsatz optimiert und unterstützt 2 verschiedene Prioritäten, sie ist durch konfigurierbare Queues realisiert, die in der Lage sind Überlastzustände abzufangen.

- Service Schnittstelle für alle Arten von Benutzer-Interaktionen. Sie werden immer mit hoher Priorität betrieben. Die Interaktionen an der Service Schnittstelle können als Systemmeldungen dokumentiert werden.

Das Ergebnis der Verarbeitung von PDM ist :

- Der Eintrag der Daten im Prozessabbild (im verteilten RTAPLS Objektmodell)
- Optional (konfigurationsabhängig)
  - Weitergabe an konfigurierbare „Kanäle“. Sie bilden auch die Schnittstelle für den Historian. Jeder dieser Kanäle kann von beliebig vielen Prozessen gelesen werden.
  - Ausgabe von Output PVs.
  - Anstoß von Berechnungen der abgeleiteten Werte

In typischen RTAPLS Konfigurationen beträgt die maximale Daueränderungsrate ca. 40.000 Prozessdatenänderungen pro Sekunde.

2.5 Der Formelinterpreter (RTAPLS - CALC)

RTAPLS – CALC wird für Berechnungen verwendet. Im Wesentlichen können damit abgeleitete Werte gebildet werden oder auch andere Operationen durchgeführt werden. Scheduler-Funktionen zur zeitgesteuerten Aktivierung verschiedenster Aktionen sind ebenfalls integriert.

RTAPLS – CALC wird mit einem Umfang an Standardfunktionen ausgeliefert, die durch projekt-spezifische Funktionen einfach erweitert werden können.

Die von RTAPLS – CALC verwendete Programmiersprache **RtPL** (Realtime Process Language)is eine Untermenge von BASIC. Zusätzlich sind noch spezielle Konstrukte für die spontanen und zyklischen Verarbeitungen Bestandteil von RtPL. Für anspruchsvolle Anwendungen unterstützt RtPL auch Objektorientierung und die Möglichkeit zur Bildung von Klassen inklusive einer einfachen Vererbung.

RtPL interpretierte einen binären Zwischencode und erreicht dabei eine Geschwindigkeit von ca. 600.000 Operationen pro Sekunde (auf 3 Ghz Pentium).

Folgende Datenquellen (Variable) können von RTAPLS-CALC verwendet werden :

- PVs inklusive Statusinformationen und allen statischen und dynamischen Attributen.
- Sonstige RTAPLS Objekte
- Beliebige RtDb Datensätze und RTA Objekte
- RtOOS – Objekte
- Systemobjekte (Uhrzeit, ...)
- Globale und lokale Variable

Die RTAPLS PV-Statusinformationen werden bei sämtlichen Berechnungen entsprechend mit verknüpft.

2.6 Der Historian (RTAPLS - HIST)

RTAPLS - HIST (Historian) wird verwendet, um Prozessdaten mit hoher Zeitauflösung über eine längere Zeit (also mehrere Jahre) zu speichern.

Binär- und Text-Daten werden mit einer Zeitauflösung von 100 Nanosekunden Analogwerte mit 1 Millisekunde archiviert. Grenzwertmeldungen für analoge Werte werden ähnlich wie das binäre Record - Format gespeichert. Der Analogwert selbst ist auch Bestandteil dieses Records.

Pro Analog-PV kann konfiguriert werden, ob Datenkompression nach dem Totband-Prinzip oder gemäß des sogenannten „Swinging-Door-Prinzips“ angewendet werden soll. Das Ziel dieser Algorithmen ist, „minimal abweichende“ bzw. vom Kurvenverlauf „vorhersehbare“ Datenpunkte nicht abzuspeichern und so den Speicherbedarf zu minimieren.

Der Plattenspeicherplatz wird in täglichen, wöchentlichen oder monatlichen Partitions organisiert. Diese Partitions können individuell auf und von einem beliebigen Backup Medium exportiert und importiert werden.

Lesende Programme können die Daten entweder so bekommen wie sie gespeichert sind (also jede einzelne Wertänderung) oder als akkumulierte Werte (Mittelwert, Minimum, Maximum,Varianz gemäß OPC HDA Standard) pro Zeitintervall.

Um den lesenden Zugriff über große Zeiträume zu optimieren werden zusätzlich vorakkumulierte Werte im Minutenintervall archiviert. Die Verwendung dieser vorakkumulierten Werte ist für den Clientzugriffe transparent.

Da sich Konfigurationsdaten während der Zeit verändern können, speichert der RTAPLS - HIST auch Konfigurationsdaten-Änderungen. Immer wenn auf RTAPLS - HIST Daten zugegriffen wird, werden die zugeordneten Konfigurationsdaten gesucht.

RTAPLS - HIST ist optimiert in Bezug auf das effiziente Schreiben von Prozessdaten mit stetig steigendem Zeitstempel. In normalen RTAPLS Konfigurationen beträgt die maximale Dauerlast ca. 50.000 Analogwertänderungen pro Sekunde, in spezialisierten Konfigurationen (feste Zeitraster, Schreiben von Rohwerten) sind auch wesentlich höhere Lasten möglich. Grundsätzlich ist es auch möglich, Daten im Nachhinein einzufügen, zu verändern oder sogar zu löschen.

Das Redundanzkonzept sorgt für lückenlose Archive auch bei einzelnen Rechnerausfällen.

2.7 Das Auskunfts- und Berichtssystem (RTAPLS - REPORT)

Die RTAPLS Objekte (PVs, Stationen, Benutzer, ...) und die Objekte des RTAPLS Historian bilden eine Objekthierarchie. Sämtliche Informationen die RTAPLS anbieten kann, haben einen festen Platz in dieser Hierarchie und stehen für RTAPLS REPORT zur Verfügung.

2.7.1 OLEDB Provider (Microsoft Plattform)

Der RTAPLS OLE DB Provider bildet eine Datenbank-Sicht über diese Objekte. Es gibt Tabellen von Binären PVs, Analogen PVs, Stationen usw. Die einzelnen Objekte (PVs) bilden die Records dieser Tabellen, die Attribute der Objekte bilden die Spalten der Tabellen. Der RTAPLS OLE DB Provider implementiert eine Untermenge von SQL um auf diese Daten mit Hilfe von SQL SELECT-Anweisungen zugreifen und verknüpfen zu können.

Sämtliche Softwarewerkzeuge die OLE DB Providern unterstützen, sind damit in der Lage auf alle RTAPLS Objekte und Attribute zuzugreifen. Eine typische Anwendung des OLE DB Providers ist die Erstellung von Reports mit Crystal Reports oder Microsoft Excel.

Weitere mögliche Anwendungen sind Web-Server für Intranetanwendungen und ad hoc Abfragen.

Auch der Export aus dem RTAPLS OLE DB Provider in eine Microsoft Access oder SQL Server Datenbank ist leicht möglich.

2.7.2 XML Provider (Linux Plattform)

Auf Linux Plattformen steht ein Berichtsystem zur Verfügung welches auf der Basis von XQuery die RTAPLS Datenquellen in XML Format zugänglich macht. Zur Formatierung der gelieferten XML Daten werden XSL basierende Reportgeneratoren verwendet.

2.8 Die Visualisierung (RTAPLS – VIEW)

RTAPLS – VIEW ist die Prozessvisualisierung von RTAPLS. Sie dient zur komfortablen Erstellung von Bedieneroberflächen für Leitsysteme. Dabei werden folgende Bereiche abgedeckt :

- Dynamische Anlagenbilder auf der Basis von erweiterbaren Symbolbibliotheken
- Überwiegend statische HTML Inhalte (Dokumentationen, Helpfunktion)
- Listen und Tabellen
- Dialoge

Als Vektorgraphikformat benutzt RTAPLS - VIEW den W3C Standard SVG (Scaleable Vector Graphics). Das hat den Vorteil, dass es unzählige Werkzeuge und Konverter gibt die es ermöglichen, vorhandenes Bildmaterial weiter zu verwenden. Die zugrunde liegende SVG Library ist plattformunabhängig und sowohl auf Windows als auch Linux Plattformen verfügbar.

Zusätzlich integriert RTAPLS – VIEW einen Web Browser für die Anzeige von HTML Inhalten. RTAPLS – VIEW kann in einen Webserver integriert werden und kann damit dynamische Anlagenbilder auch mit Web Technologie in Büroumgebungen verfügbar machen.

Ein Visualisierungssystem beinhaltet auch Logikabläufe die zum Beispiel den Bildwechsel steuern. Komplexere Abläufe, die projektspezifisch definiert werden müssen, benötigen eine Programmiersprache.

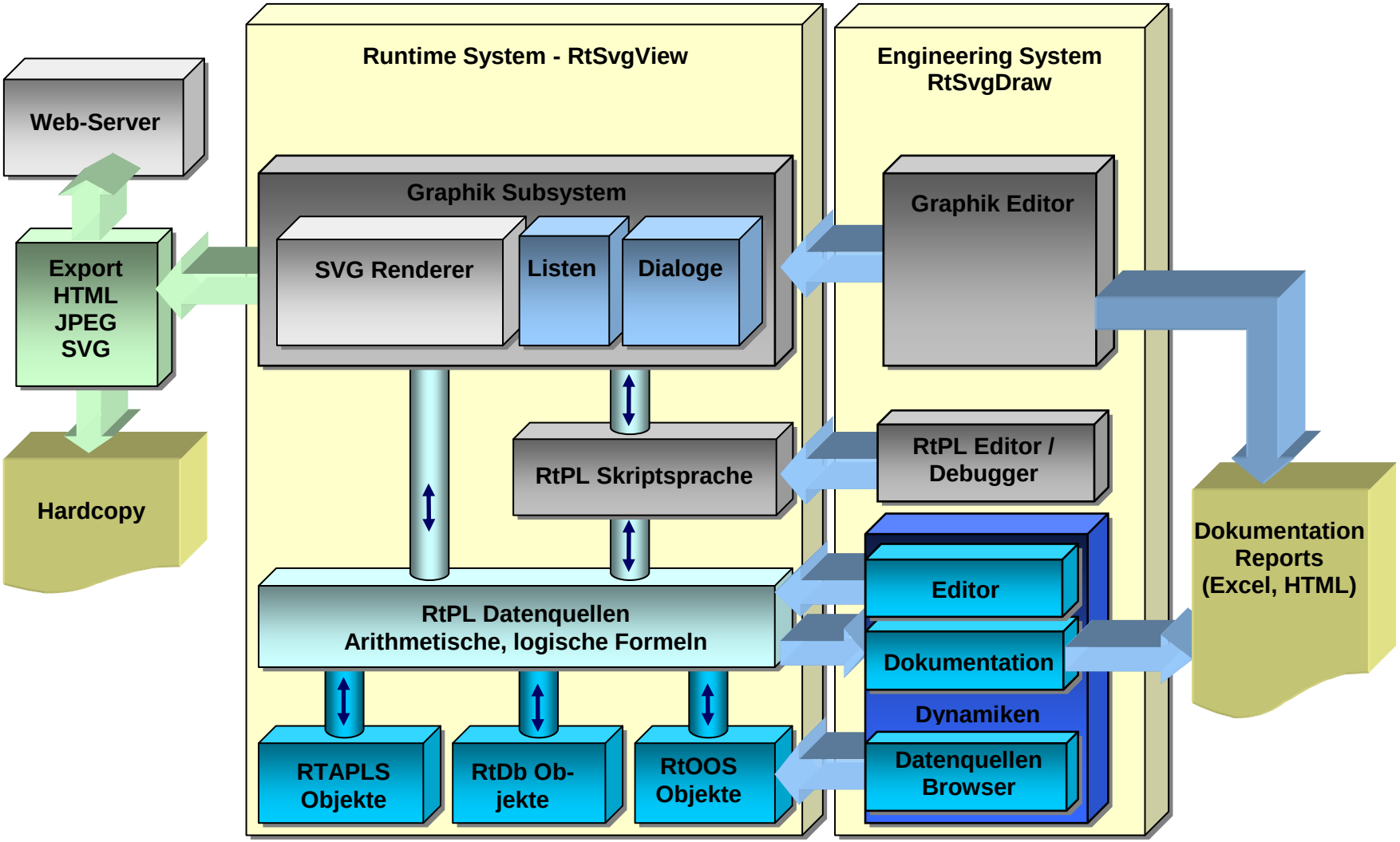
In RTAPLS - VIEW wird als Programmiersprache ***RtPL***, die Sprache von RTAPLS – CALC verwendet. Zusätzlich zu den Datenobjekten (PVs, RtDb Daten, ...) sind auch sämtliche Graphischen Objekte des Bildes in RtPL wie Variable ansprechbar. RtPL ist syntaktisch mit Visual Basic kompatibel und damit leicht erlernbar.

Die Dynamikanbindung stellt die Verbindung zwischen den Attributen des Graphiksystems und den Daten her. Als Daten kommen die gleichen Datenquellen wie bei RTAPLS-CALC in Betracht (PVs, RtDb Daten, RtOOS Objekte, ...). In der Regel müssen diese Daten jedoch verknüpft und transformiert werden (z.B muss aus einem Zustand eine Farbe werden). Zu diesem Zweck unterstützt die Dynamikanbindung gleiche arithmetisch / logische Verknüpfungen wie RTAPLS-CALC aber auch einfach zu erstellende Transformationstabellen.

Die Dynamikanbindung (Verknüpfungen) können wahlweise auch aus der Visualisierung in einen eigenen RTAPLS-CALC ausgelagert werden, der diese Verknüpfung für alle Visualisierungs-Workstation gemeinsam ausführt. Die Schnittstelle zu den Visualisierungs Dynamiken werden in dem Fall durch Visualisierungs-Variable, ein Art von „leichtgewichtigen“ PV realisiert.

Eine IEC 1772 konforme Symbolbibliothek ist in RTAPLS - VIEW enthalten.

2.8.1 RTAPLS VIEW - Architektur



2.9 SOA / Web Server (RTAPLS – SOA)

Der RTAPLS SOA / Web Server ermöglicht die Integration von RTAPLS in die gesamte unternehmensweite Datenverarbeitung beziehungsweise in Büroumgebungen basierend auf der Service Oriented Architecture (SOA). Möglich ist das durch die von W3C standardisierten Web Services. Diese Protokolle auf der Basis von XML ermöglichen den Zugriff auf alle RTAPLS Objekte in einer vollkommen plattformunabhängigen Weise auch mit dem „firewall freundlichen“ HTTP protokoll.

Der RTAPLS SOA / Web Server integriert mehrere Funktionalitäten.

- RTAPLS Web Server
  - HTML Inhalte
    - Help system
    - Online RTA / RTAPLS Statusübersicht
  - XML Inhalte
    - RTA / RTAPLS Objekte (werden im XML Format geliefert und mit XSLT Stylesheets entsprechend aufbereitet).
    - RTAPLS Konfigurationsdaten
    - RTA Statusübersicht

- RTAPLS SOA Server (HTTP basierend)
  - RTA und RTAPLS spezifische, W3C konforme Web Services
  - Genormter Prozessdatenzugriff OPC Unified Architecture Data Access 1.0

- RTAPLS RPC Remote Server (TCP/IP)
  - Binärer Zugriff auf alle RTA und RTAPLS Objekte

2.10 Objektorientiertes Engineering (RTOOS)

Objektorientierung und Klassenbildung sind die nötigen Mechanismen um die Engineeringaufwände in umfangreichen Projekten niedrig zu halten und um Prozesse zu modellieren. Einen Prozess bzw. eine Anlage nur als homogene Menge von zigtausend Datenpunkten zu sehen ist unübersichtlich, fehleranfällig und mühsam. Tatsächlich orientiert sich das menschliche Denken gerne und erfolgreich an Hierarchien.

Das wesentliche an RtOOS ist, dass eine Anlage durch hierarchische Klassen beschrieben wird. Klassen sind abstrakte Sammelbegriffe für Objekte die „weitgehend" gleichartig sind. Das Verhalten dieser Objekte wird in der Klasse beschreiben (= ***Methoden***). Ein Objekt verfügt über statische und dynamische ***Attribute***, die seinen Zustand beschreiben. Dynamische Attribute werden durch **Trigger** (spontan oder zyklisch durchzuführende Berechnungsvorschriften) berechnet. Zwischen den Objekten bestehen ***Relationen***. Einige der ***Relationen*** beschreiben die Objekthierarchie (ein Objekt ist *Teil eines anderen* Objekts, oder ein Objekt *enthält* Unterobjekte). Die unterste Ebene dieser Hierarchie bilden typischerweise die PVs, die ja auch Objekte sind, mit bereits vordefinierten Methoden und Attributen.

Zur Realisierung der Klassen und Objekte :

- Die Objektattribute und die Relationen werden durch RtDb-Tabellen realisiert.
- Die Objekte können dadurch Server- oder Client seitig gebildet werden
- Die Trigger und Methoden der Klassen werden mit ***RtPL*** oder **C++** definiert.
- Engineeringwerkzeuge unterstützen Generierung und Test der Klassen

2.11 Die System Administration (RTAPLS – CONSOLE)

RTAPLS enthält ein umfangreiches Administrations-Konsole mit folgenden Funktionen :

- Error Logging
  Sämtliche Prozesse produzieren Einträge in ein zentrales Log File. Dieses Logfile wird stündlich neu angelegt. Bei der Windows Version wird zusätzlich im Windows Event Log mitgeschrieben. Der OLEDB Provider ermöglicht den flexiblen Zugriff auf das Log File. Ein Error-Eintrag enthält
  - Zeitpunkt
  - Kategorie (Information, Warnung, Fataler Fehler)
  - Verursacher (Programmname, Modul)
  - Fehlercode
  - Text mit dynamischen Informationen

- Trace System
  Das Trace System steuert die Ausgabe von Trace-Informationen in verschiedenen Prozessen. In den Treibern und in RTAPLS PDM kann z.B. bis auf PV Level getraced werden um die Verarbeitung einzelner PVs zu verfolgen.

- Task und Netzwerk Monitoring
  RTA verwaltet Informationen über die Programme (Fehler, Zeitpunkt des letzten und ersten Fehlers, ...). Auch wichtige Resourcen wie Queues und Real Time Database werden ständig beobachtet.

- Statistiken
  Programme wie PDM und HIST bilden statistische Informationen über Datendurchsatz und Änderungshäufigkeiten teilweise bis auf PV Level. Diese Daten können z.B. zur Optimierung der Deadbands herangezogen werden.

- Prozessdaten - Explorer
  Über eine Baumstruktur können die Prozessdaten (PVs, Anlagenbereiche, RtOOS Objekte) inspiziert und, entsprechende Benutzerrechte vorausgesetzt, auch manipuliert werden (Signalkontrolle, Sperre, manuelle Wertvorgabe, …).

- Benutzerverwaltung
  Verwaltung der Rechte pro Benutzergruppe (=Rolle) und der Benutzer selbst

- Historian
  Zugriff auf sämtliche Prozessdaten im Historian in Listenform und als Trendgrafik. Partition Management (Import, Export, Archivieren).

- Online Signaldarstellung
  Statische oder animierte Signallisten aufgrund von Selektionskriterien :
  - Statisch : AKZ, Signaltext, Beschreibung, Anlagenteil, PLC Adresse, Priorität
  - Dynamisch : Wert, Signalqualität, Alarmzustand, Sperren

  Sortierungen : AKZ, Feldzeit, Text, Signalnummer, Priorität
  Durch definierbare PV Gruppen gesteuerte Darstellungen
  - Messwert - Balken
  - Messwert - Kurven (Linie, Balken, …)
  - Mosaikdarstellung (kompakte Darstellung für Binärdaten)

- PV Gruppen Verwaltung

3 Appendix A – Screenshots RTA/RTAPLS Console

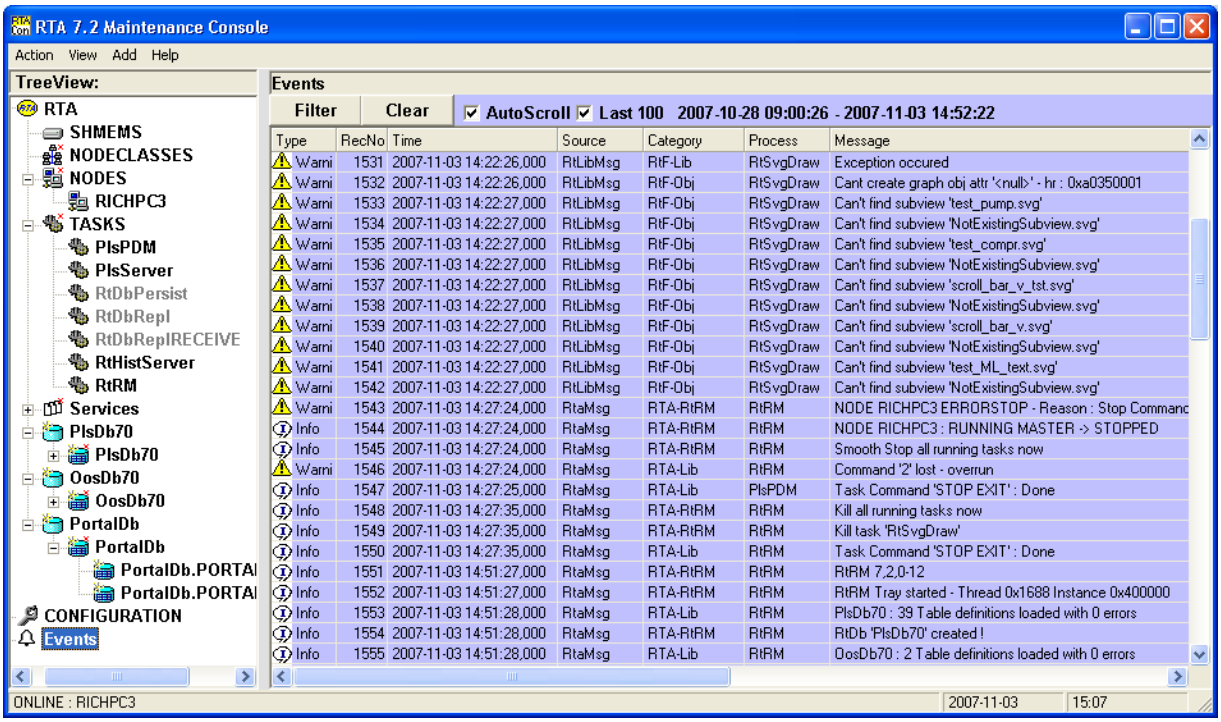


Abbildung 3.1 : RTA CONSOLE – Event log

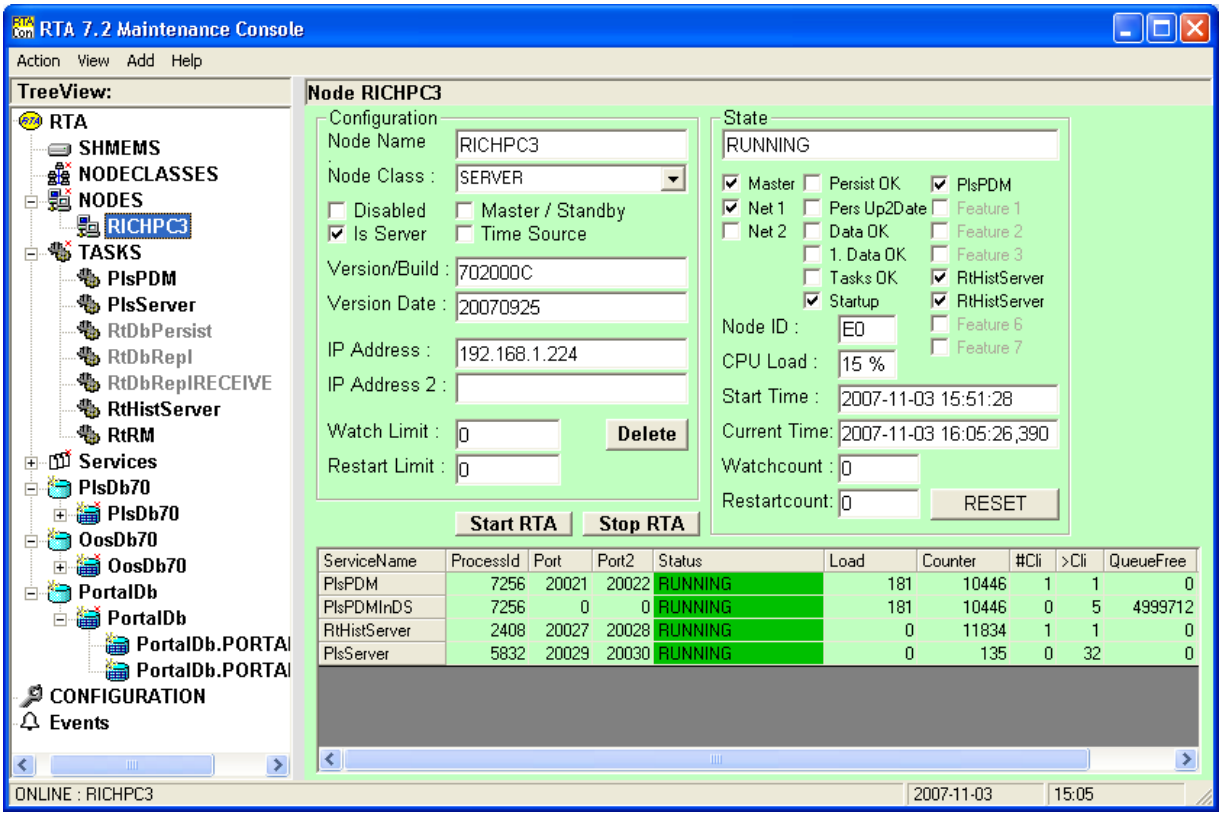


Abbildung 3.2 : RTA CONSOLE – Node Monitoring

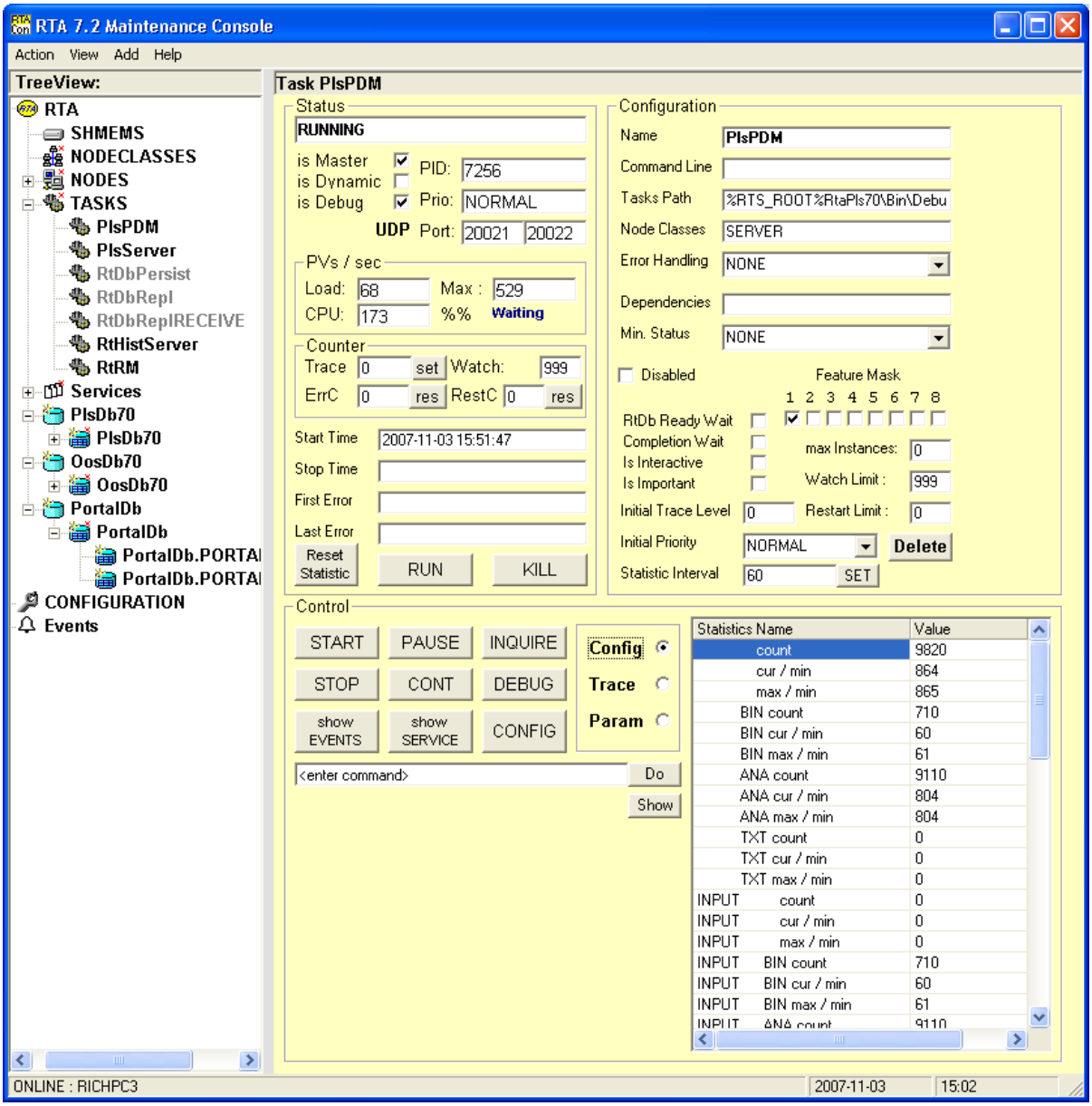


Abbildung 3.3 : RTA CONSOLE – Task Monitoring

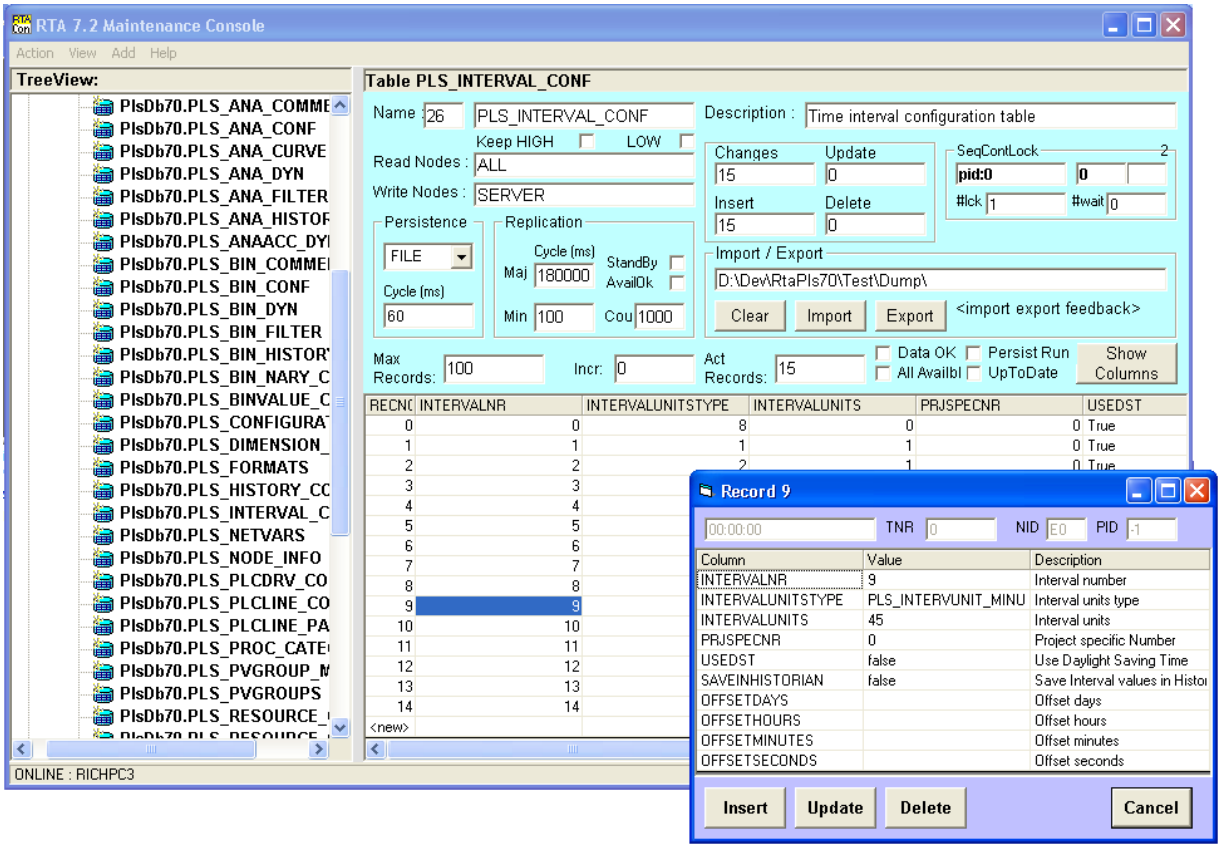


Abbildung 3.4 : RTA CONSOLE – RtDb Editor

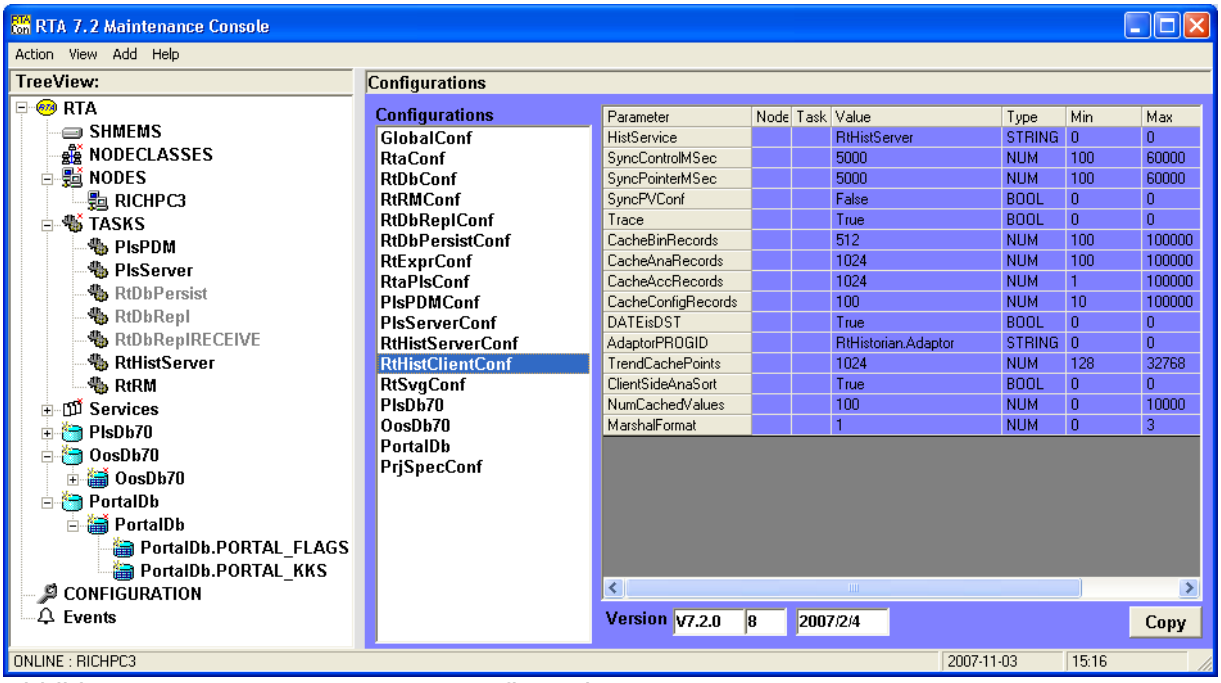


Abbildung 3.5 : RTA CONSOLE – Konfigurationsparameter

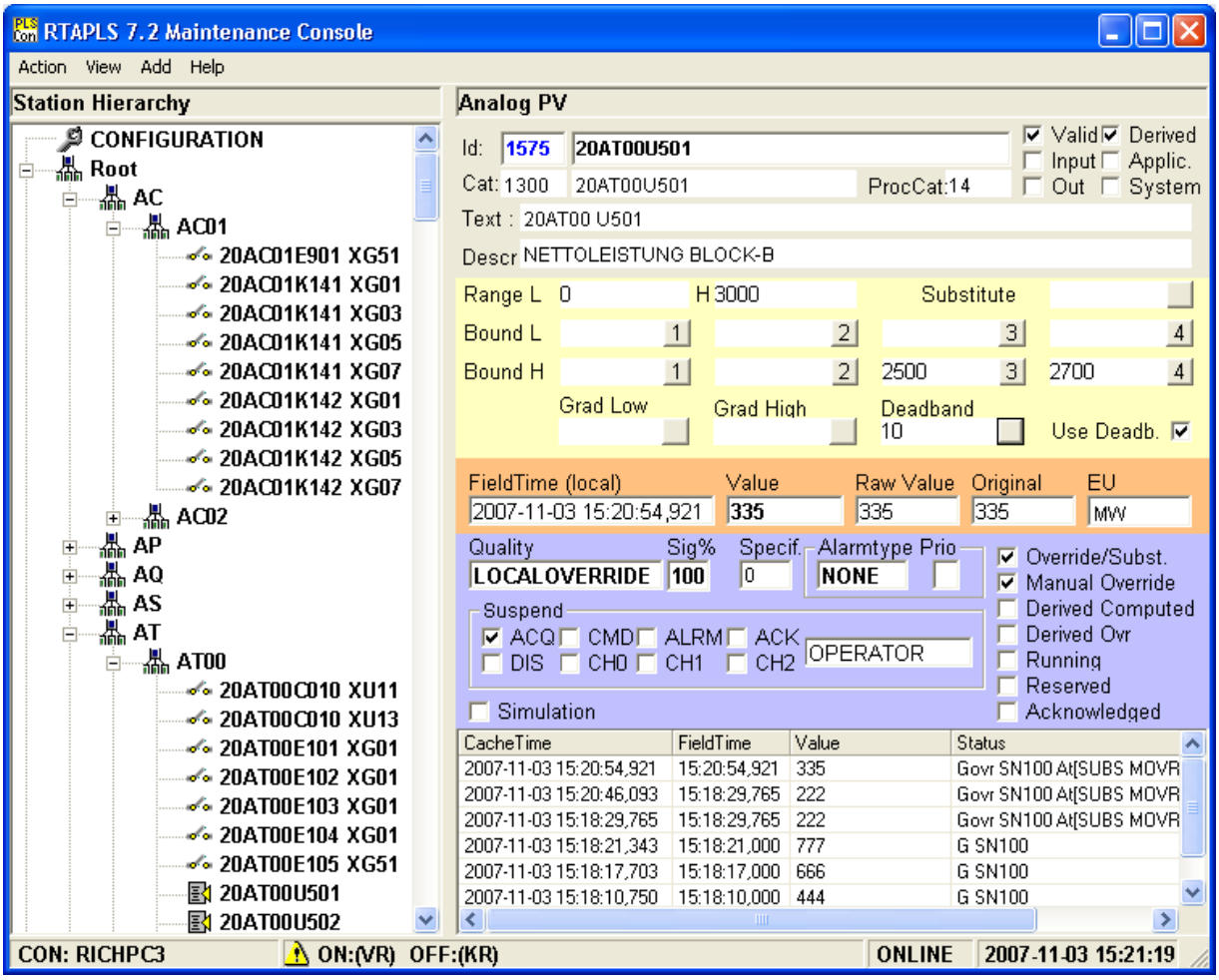


Abbildung 3.6 : RTAPLS CONSOLE – Anlagenhierarchie / Prozessvariable

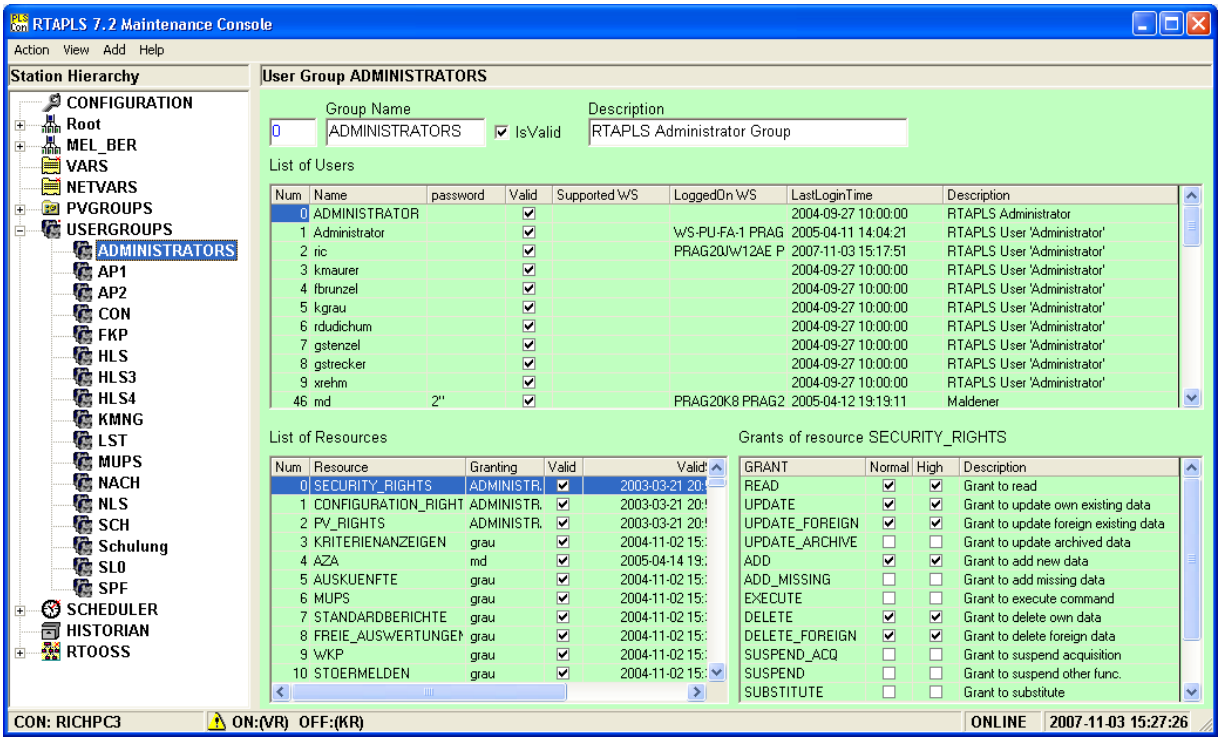


Abbildung 3.7 : RTAPLS CONSOLE – Berechtigungen

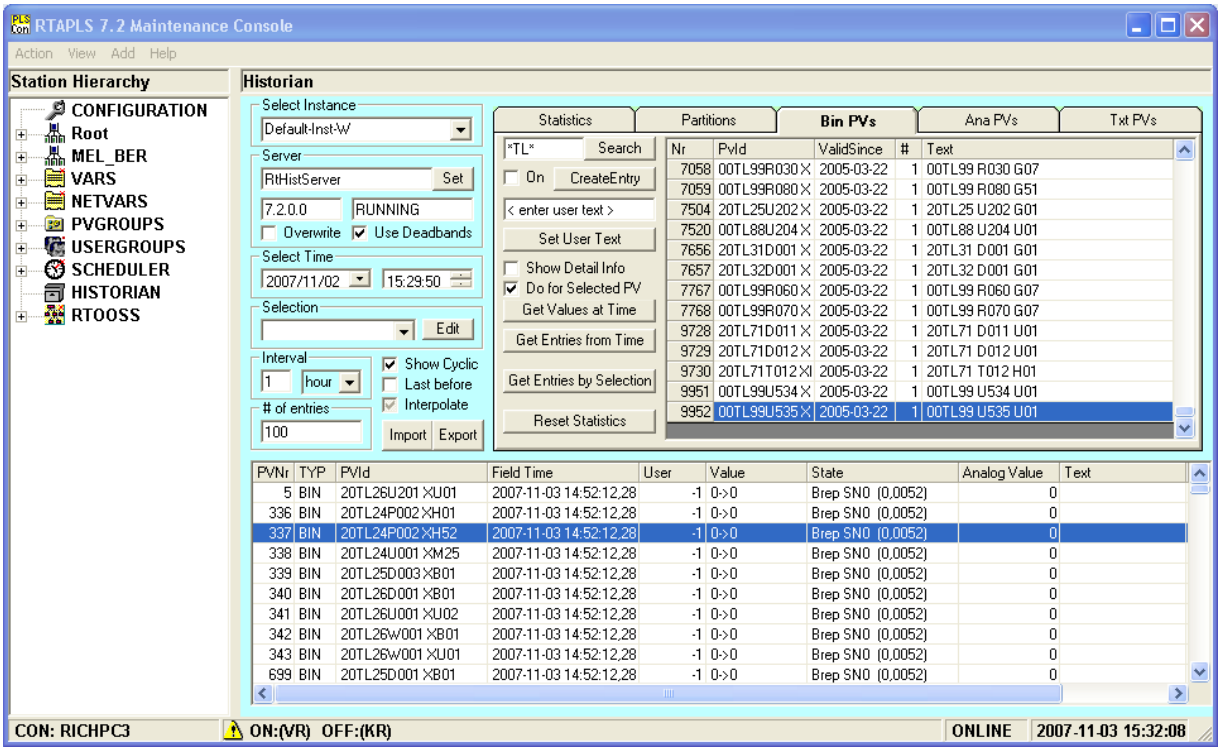


Abbildung 3.8 : RTAPLS CONSOLE – Historian Zugriff

4 Appendix B – Screenshots RTSVG Editor

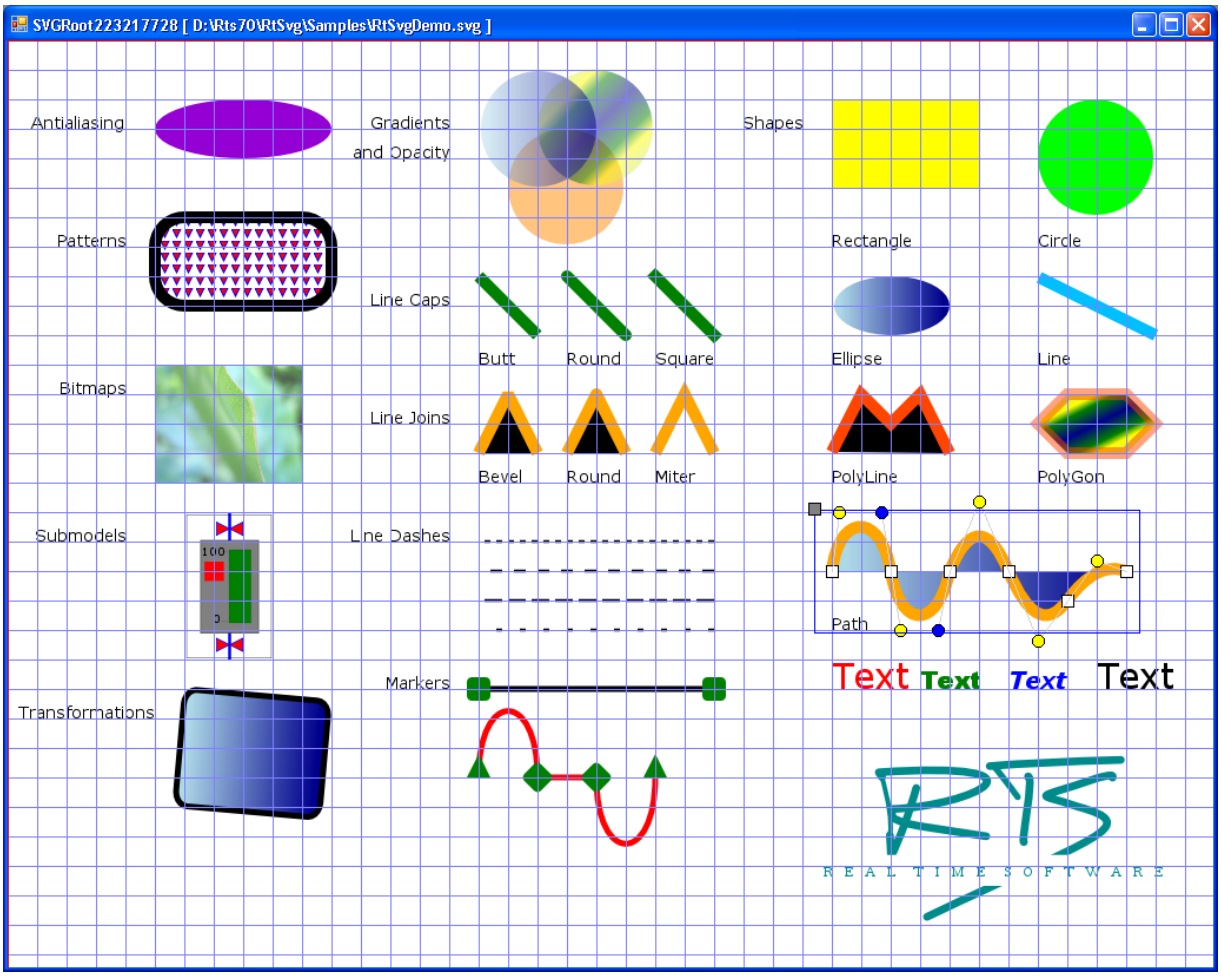


Abbildung 4.1 : RtSvgDraw – graphische Elemente

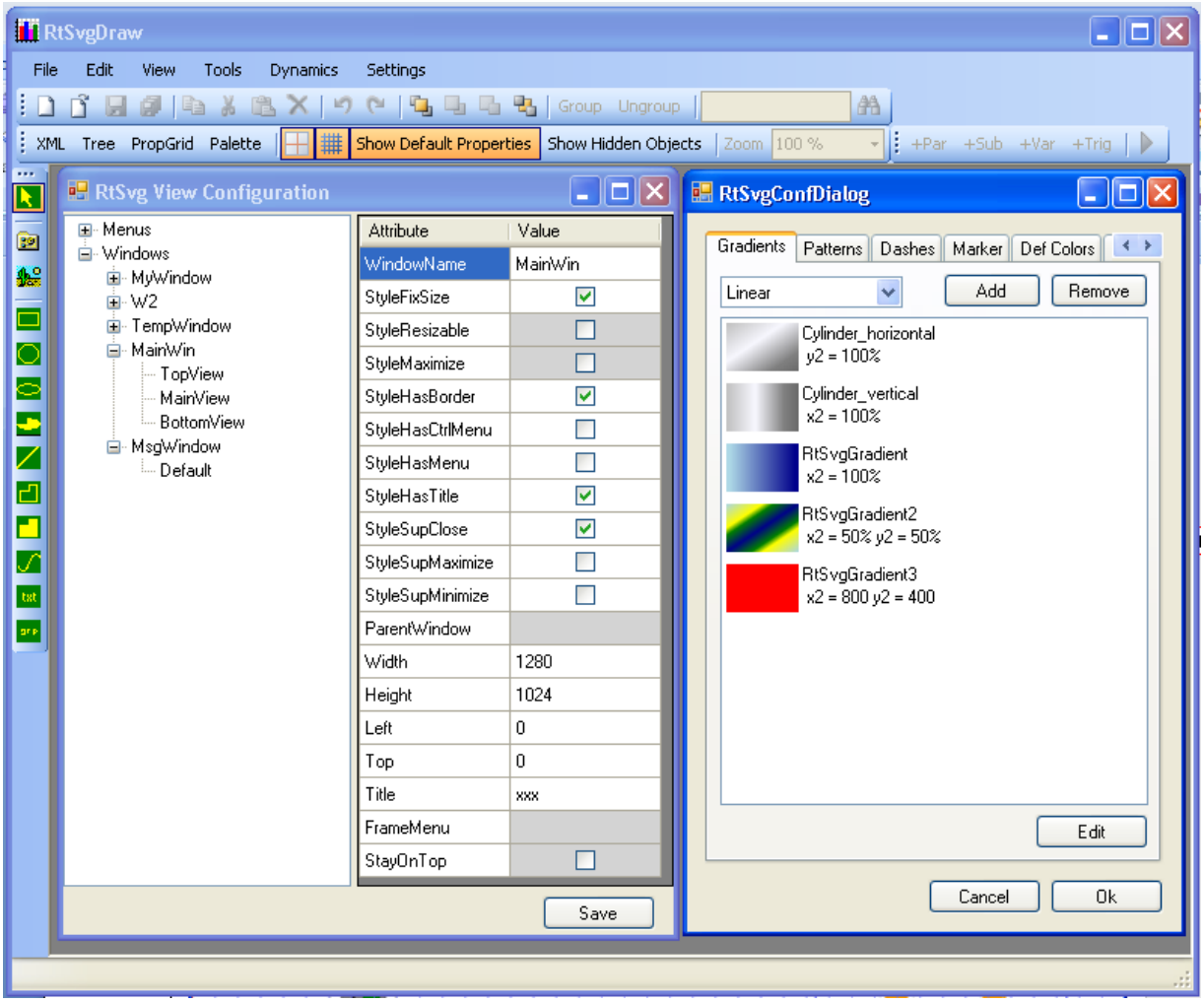


Abbildung 4.2 : RtSvgDraw– Projekteinstellungen

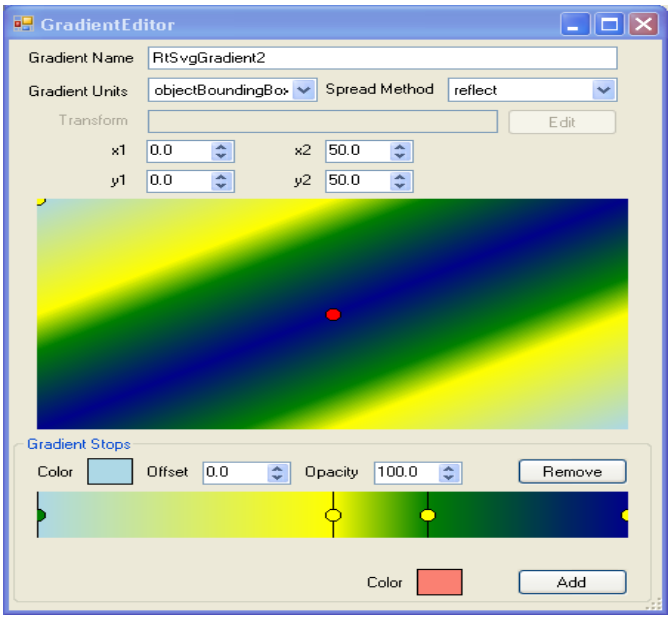


Abbildung 4.3 : RtSvgDraw : Gradienteneditor

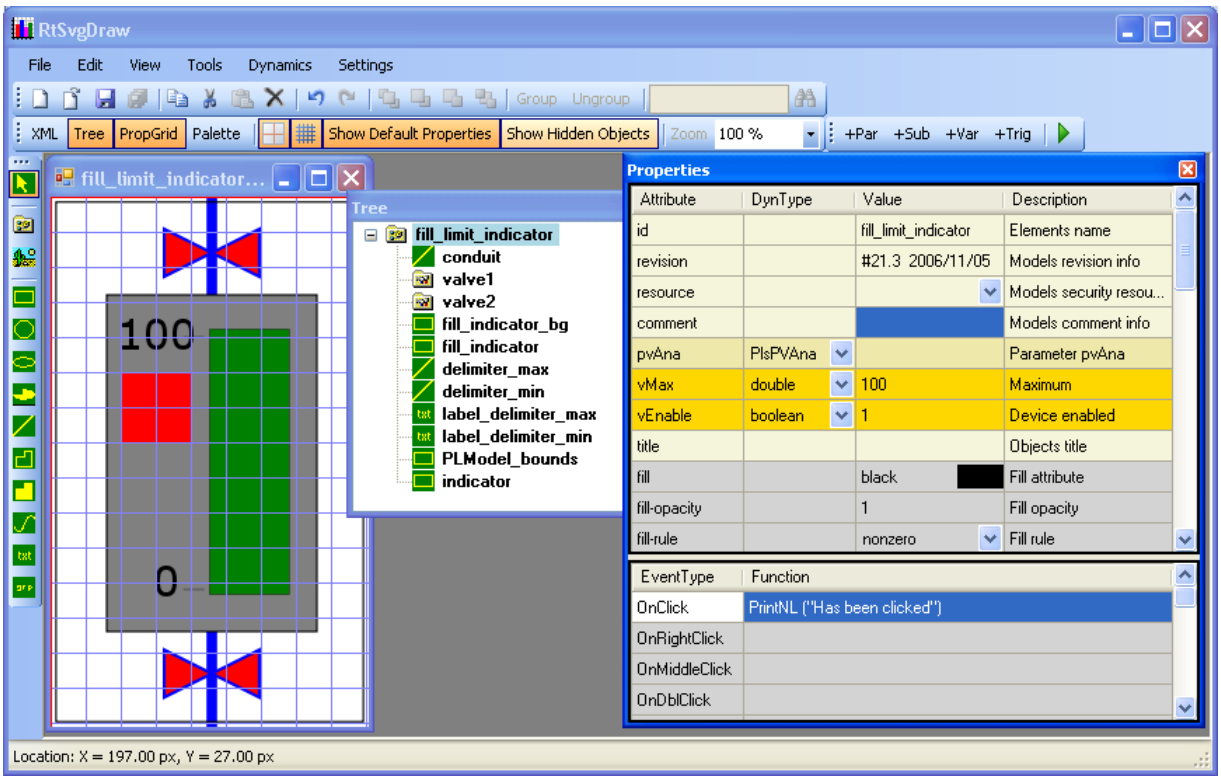


Abbildung 4.4 : RtSvgDraw – Subview Definition

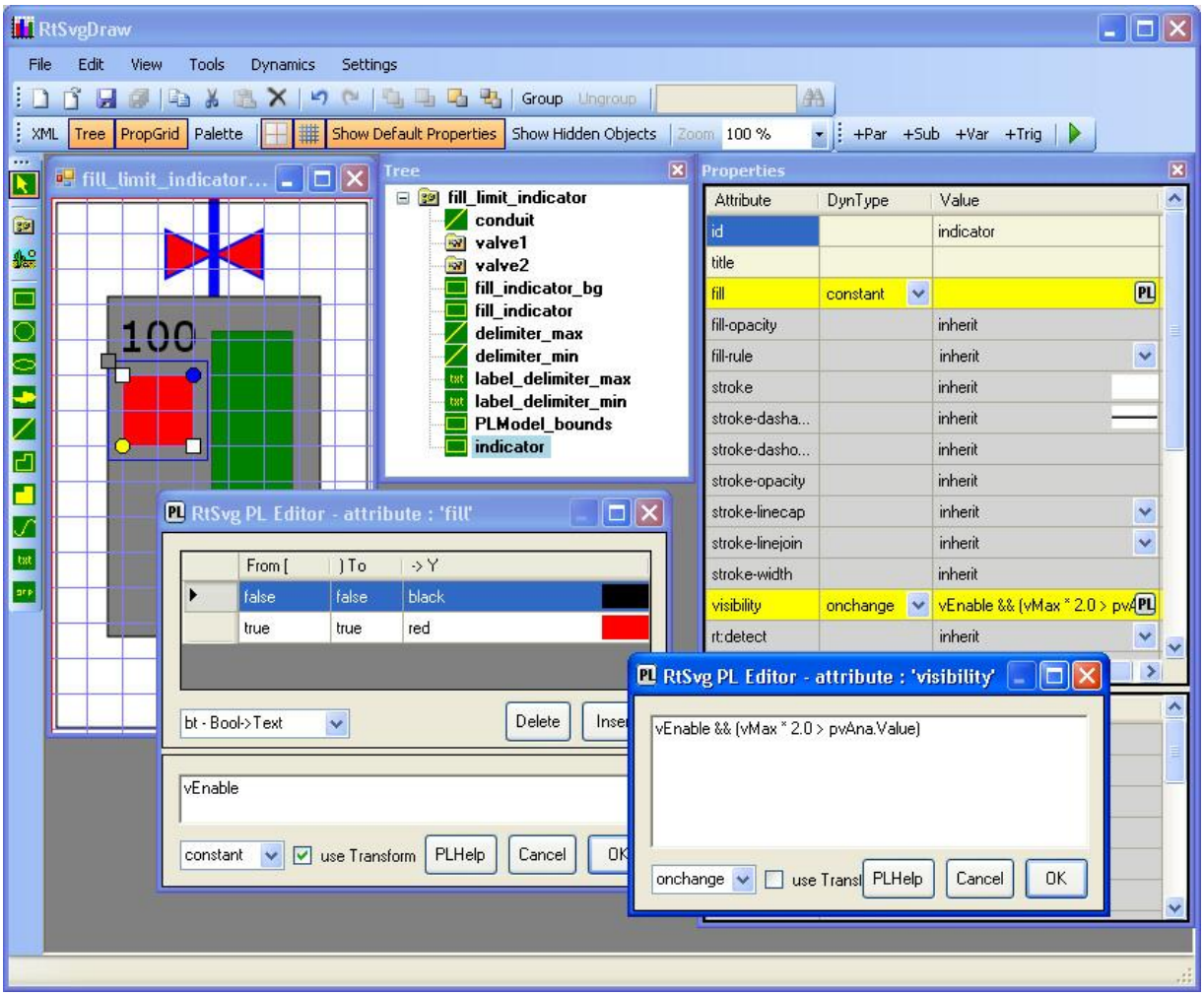


Abbildung 4.5 : RtSvgDraw – Subview Dynamiken

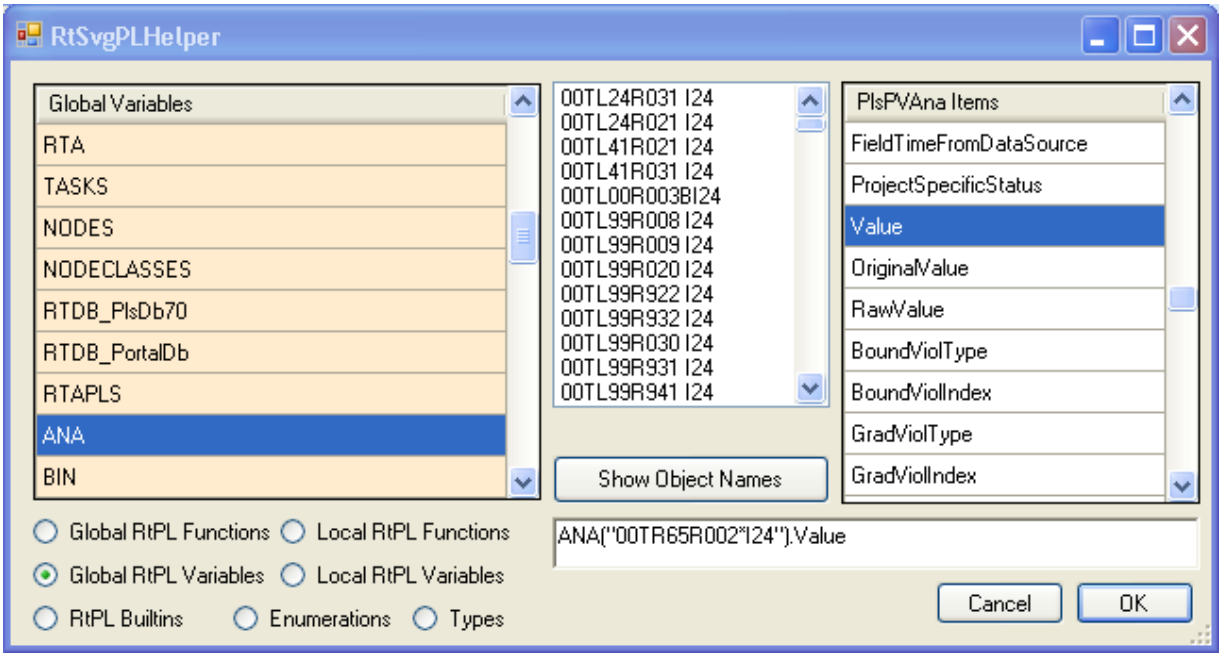


Abbildung 4.6 : RtDvgDraw – RtPL Datenquellen Dialog

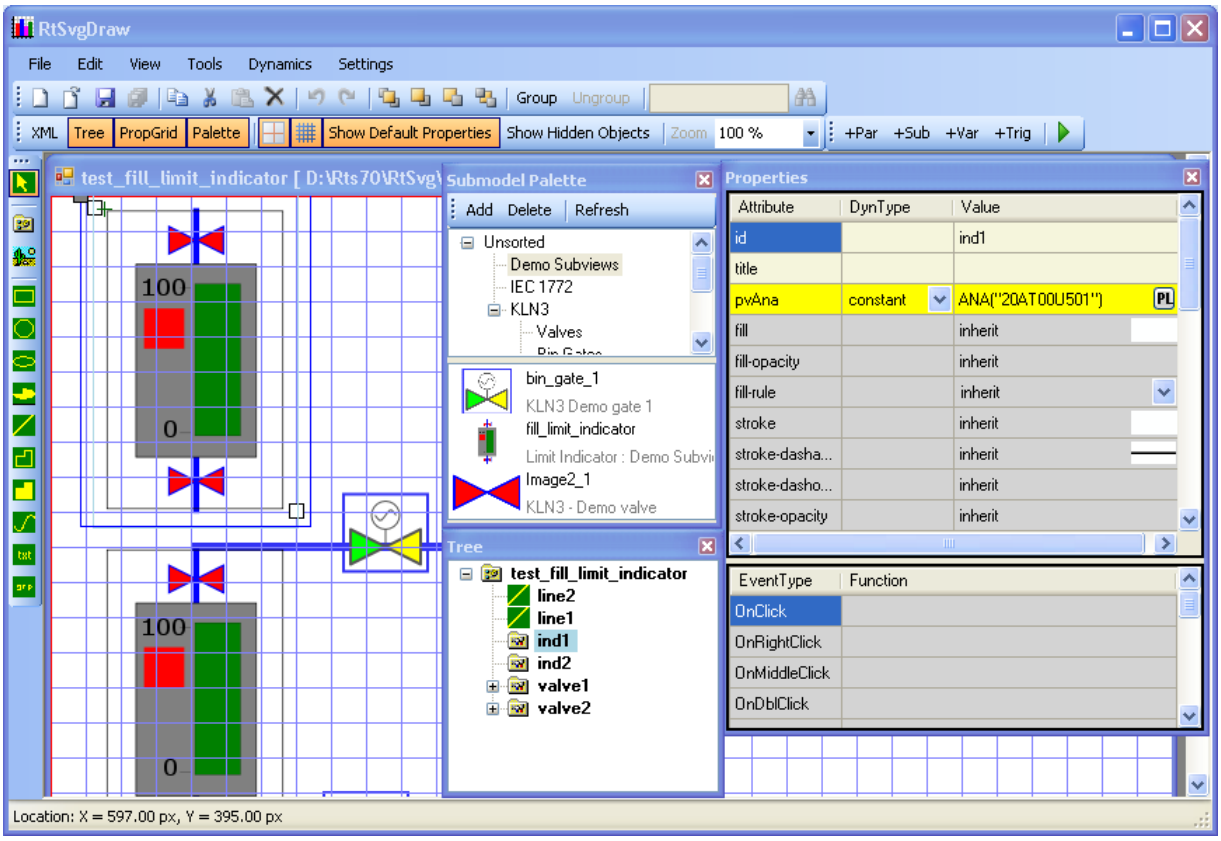


Abbildung 4.7 : RtSvgDraw – Subview Verwendung

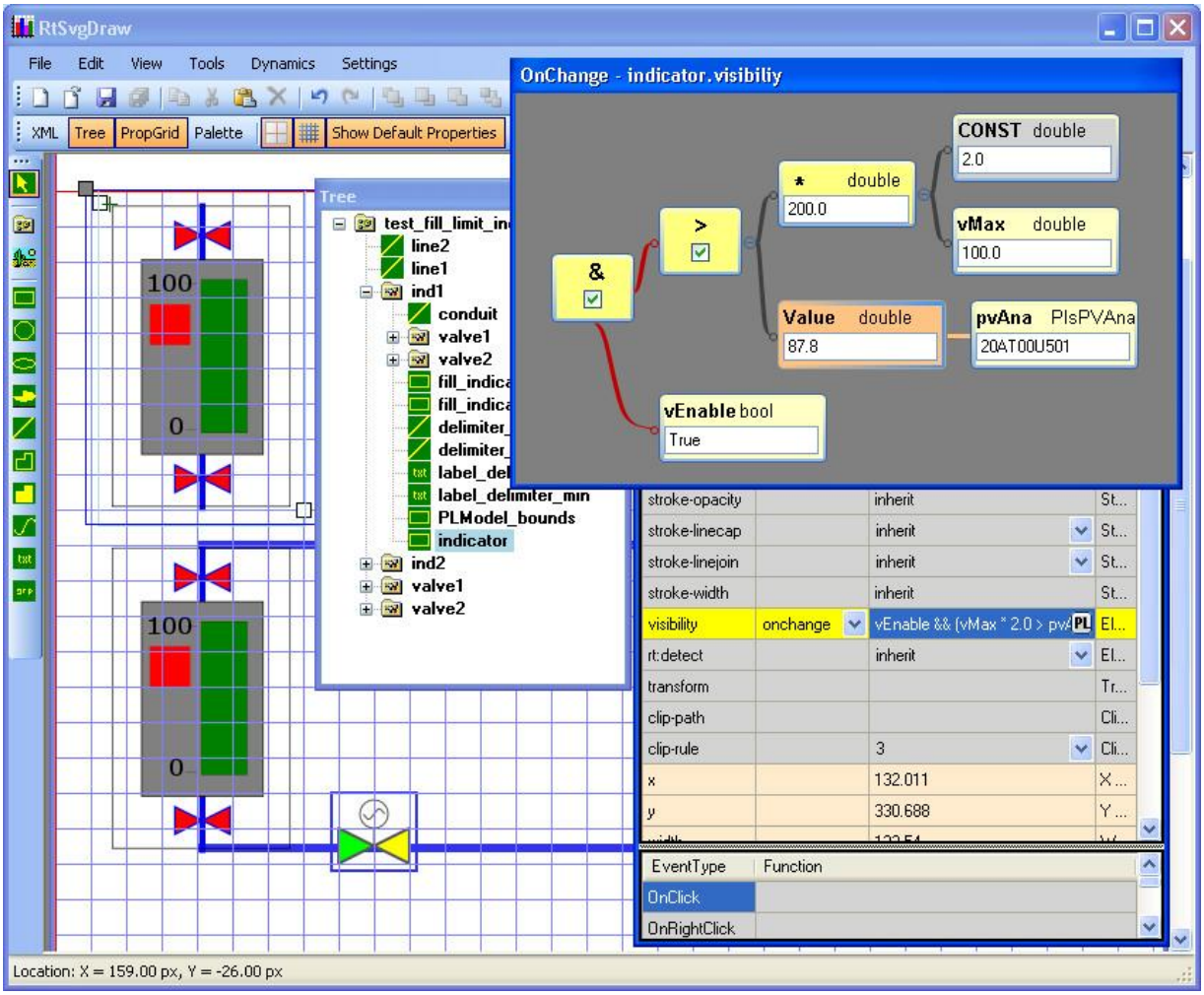


Abbildung 4.8 : RtSvgDraw - Model Test

5 Appendix C – Screenshots von Projekten

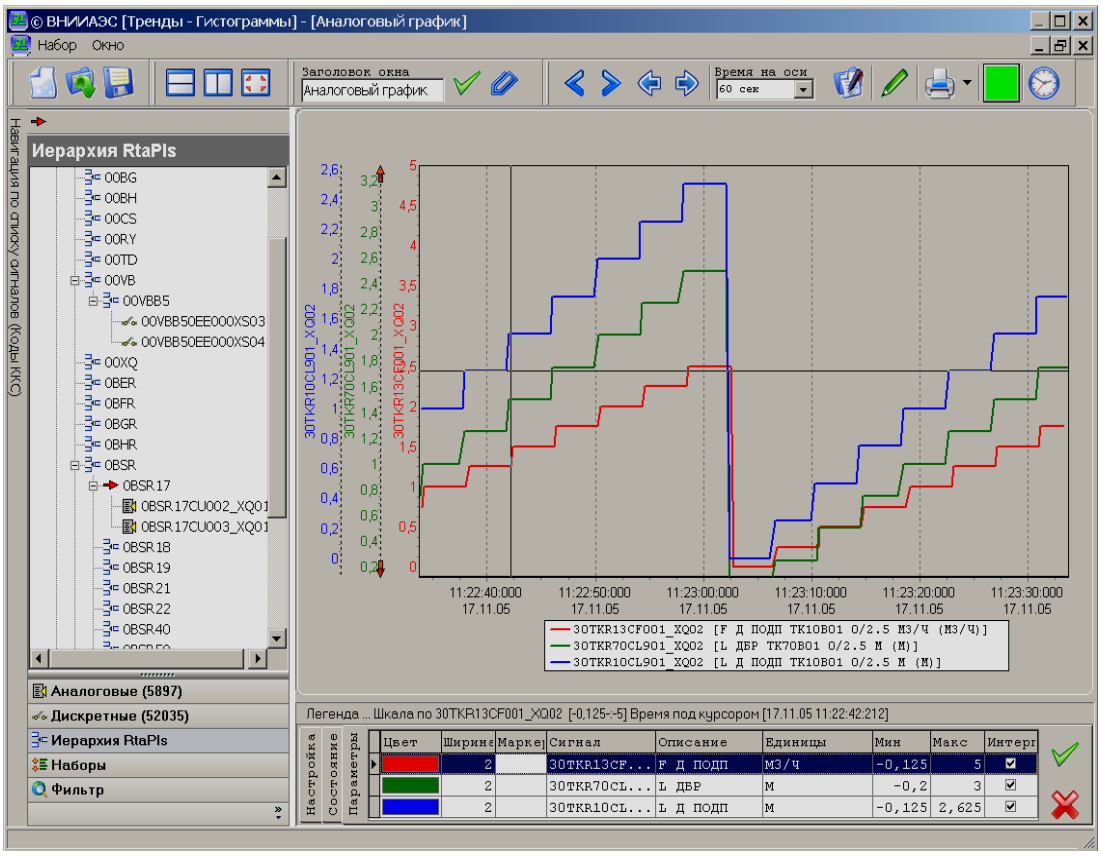


Abbildung 5.1 : Trendbilder Historian

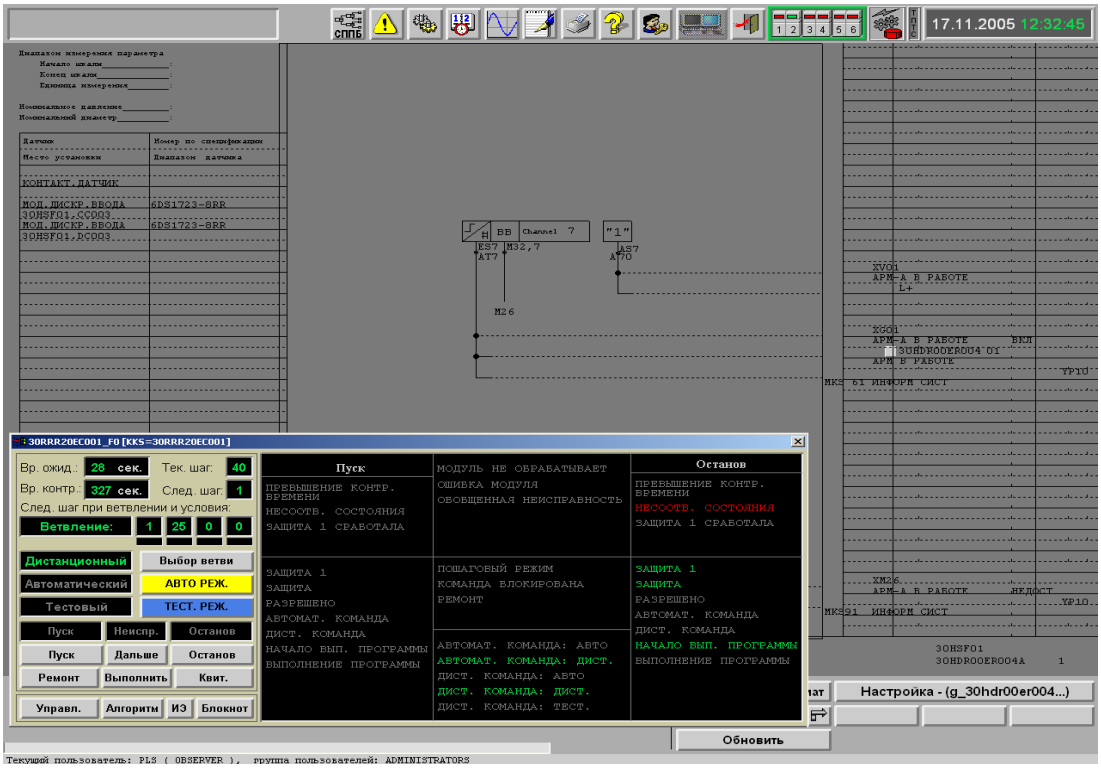


Abbildung 5.2 : Kriterienbild – automatisch generiert

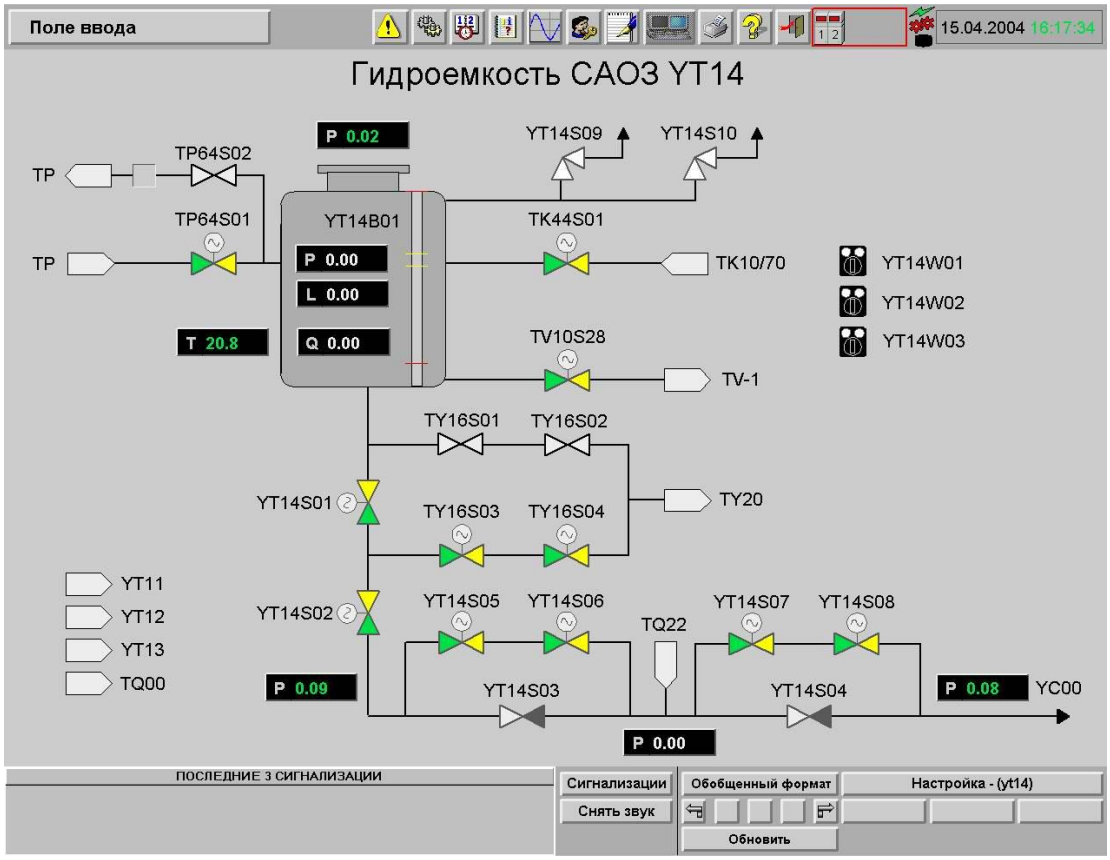


Abbildung 5.3 : Anlagenbild

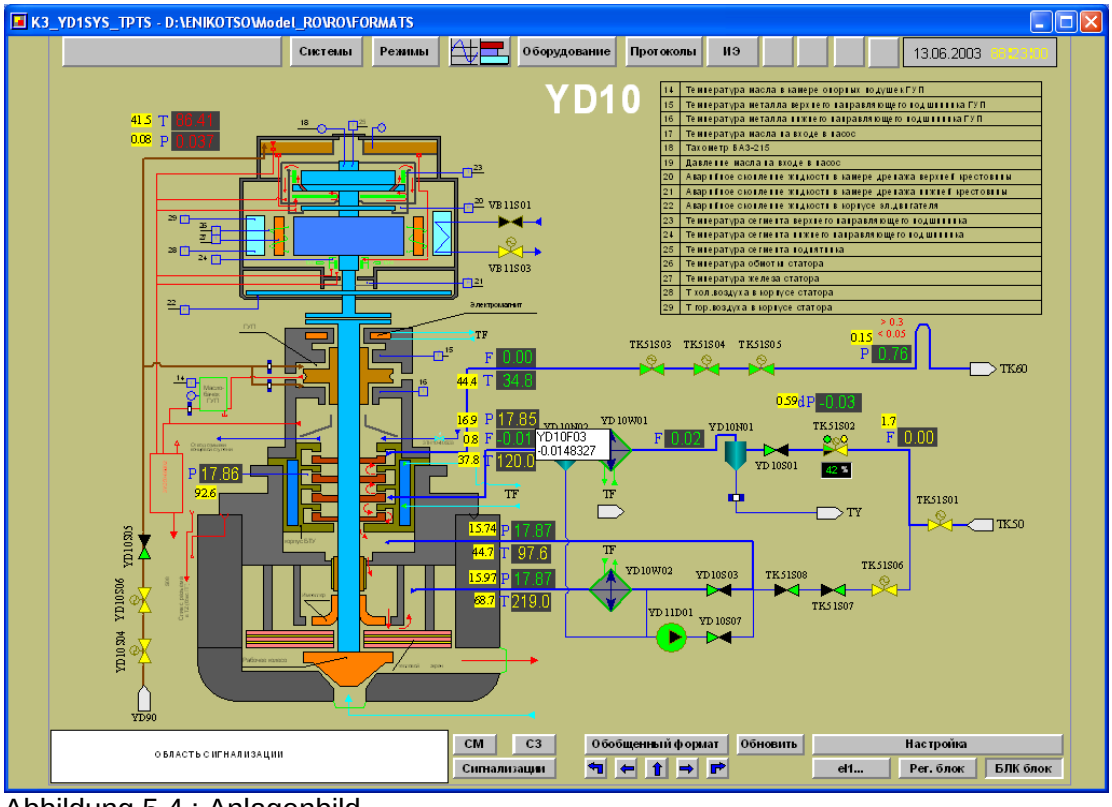


Abbildung 5.4 : Anlagenbild

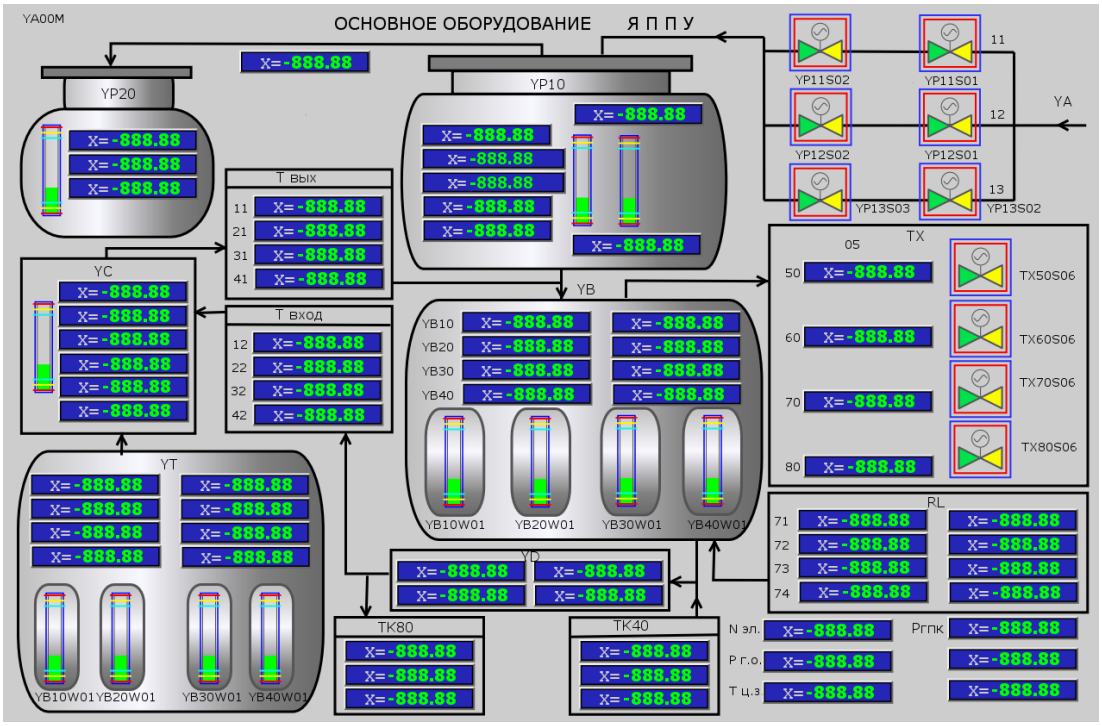


Abbildung 5.5 : Anlagenbild

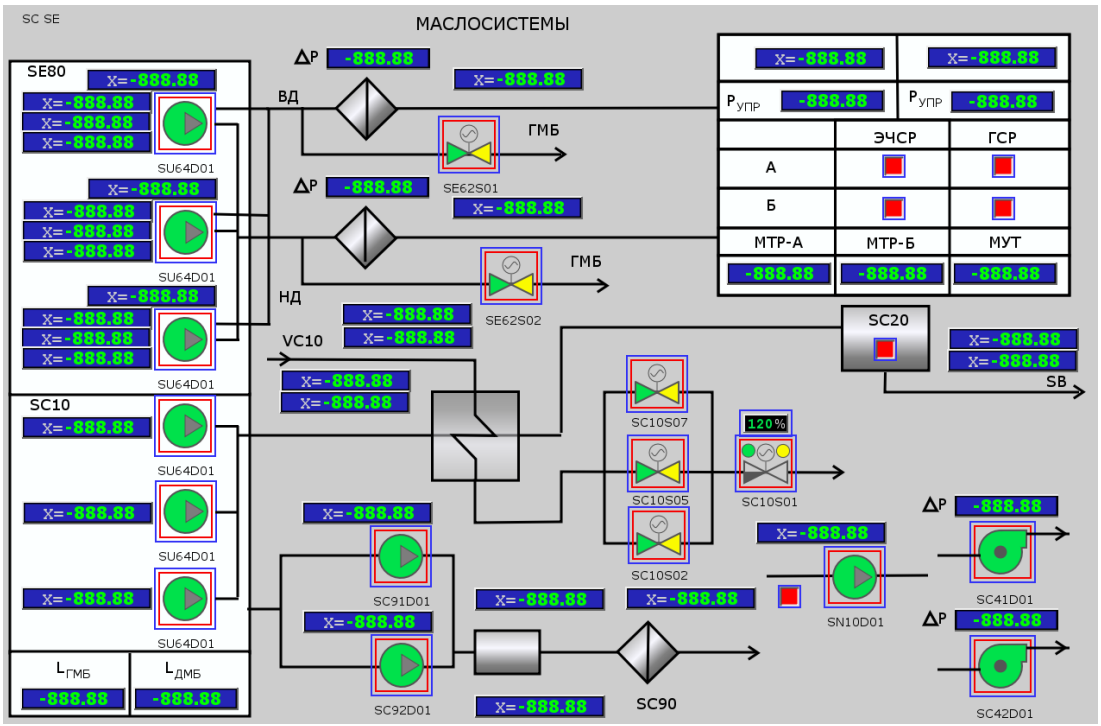


Abbildung 5.6 : Anlagenbild