ON DESIGNING EXTENDABLE JAVA-BASED SOFTWARE FOR TIME SERIES ANALYSIS

ALBEANU, Grigore
Faculty of Mathematics and Informatics
Spiru Haret University
g.albeanu.mi@spiruharet.ro

MADSEN, Henrik
DTU Informatics, Lungby 2800, Denmark
hm@imm.dtu.dk

POPENŢIU-VLĂDICESCU, Florin
University of Oradea, UNESCO Chair in Information Technologies &
City University London, UNESCO Chair in Information and Communication
popentiu@imm.dtu.dk

DUMITRU, Marius
Spiru Haret University, Faculty of Mathematics and Informatics &
IT Smart Systems, 26 Tibles Street, Bucharest, Romania
marius_dumitru86@yahoo.com

Abstract

Time series analysis is an important topic both for teaching statistics and research. The time series approach is used to predict future values based on the history of past values. Previous results in time series analysis methodologies, relevant applications and software development for time series analysis motivate us to promote a recently developed extendable Java-based tool for time series analysis. The software design methodology, the implemented algorithms and the extension methodology are detailed.

Keywords: *time series, extendable software, component-based development, Java*

ACM Classification: K.6.3, G.4

1. Introduction

Discovering knowledge by processing time series is an important objective for effective management of a large class of phenomena. Stationary and non-stationary time series models are used in various fields. There are two

main approaches to time series analysis. The first approach (*time domain*) represents time series as a function of time and is used to obtain the trend component and, then, to propose a prediction model. The second approach deals with the *frequency domain*, to determine the periodic components of the series, according to (Hamilton 1994) and (Madsen et al. 2006). Other aspects a related to time series segmentation, times series similarity, pattern discovery etc.

Several methodologies have been proposed and used to analyse and forecast time series ranging from linear modelling to non-linear techniques including those based on neural networks (Gheyas & Smith 2009). Also fuzzy modelling (Hong 2005) received great interest and useful results were obtained in some fields of application.

Previous results in time series analysis methodologies, relevant applications and software development for time series analysis [2, 3, 4, 13, 16] motivate us to promote a recently developed extendable Java-based tool for time series analysis. This paper describes SADM - a Java-based software tool for time series management, analysis and visual description. The most important attribute of the software is the extendability. By design, new methods of investigation will be added and others will be updated easily.

Only basic methods for time series analysis were implemented till now, but more methods will be connected in future.

The software can be used not only as a software engineering demonstration tool, but also for experiments and research in the field of applied time series like financial, stock exchange, signal processing etc.

2. **Time series basic algorithmic methods**

Let the value of the time series at some time point t be denoted by $x[t]$. The observed values can be represented as a sequence having $x[0]$ - the initial time point, $x[1]$ - the second time point and so forth out to $x[n-1]$ - the last observed value, n being the observed values.

A stationary time series is one for which the statistical behaviour of $x[t_1], x[t_2], \ldots, x[t_k]$ is identical to that of the shifted set $x[t_1+h]$, $x[t_2+h]$,..., $x[t_k+h]$ for any sequence of time points $t_1, t_2, \ldots, t_k$ and for any h the *lag* or the length of the interval between two sets of observations.

Stationarity of time series implies the homogeneity of the time series; that means the series behaves in a similar way regardless the time sampling. Mathematically, the stationarity implies the invariance of the joint probability distribution of the process under observation, as explained in (Madsen et al. 2006).

Weak stationarity is used in many cases, asking for constant mean over time: $E(x[t]) = \mu$, and the population *covariance function* will depend only on h: $E((x[t+h]-\mu)(x[t]-\mu)) = \gamma_x(h)$. E is the averaging operator over the population densities and h is the lag.

The scaled version of the covariance function is called ACF *the Auto-Corelation Function*, and is given by $\rho_x(h) = \gamma_x(h)/\gamma_x(0)$, with values between

-1 and 1, being useful to establish the correlation at adjacent points of the same series. Many time series analysing techniques are based on the idea that a suitably modified time series can be regarded as weakly stationary.

The most used methods for time series modification are: *detrending* (based on some parametric model and analyse the residual series), *differencing* (by first order or high order difference operation), applying different *transformations* (logarithmic, square root, Box-Cox etc.), and *filtering* (various linear combinations depending on the field of application).

Using graphical methods to analyse a given time series an indication of possible linear as well as nonlinear relations can be obtained based on the *lagged scatter plots* (the plots that put $x[t]$ on the horizontal axis and $x[t+h]$ on the vertical axis) using various values of the lag $h = 1, 2, 3, \ldots, m$.

The measuring of the ACF is based on the following algorithm:

$$\bar{x} = \frac{1}{n}\sum_{t=0}^{n-1} x[t], \bar{y} = \frac{1}{n}\sum_{t=0}^{n-1} y[t],$$

$$\hat{\gamma}_x(h) = \frac{1}{n}\sum_{t=0}^{n-h-1} (x[t+h] - \bar{x})(x[t] - \bar{x}),$$

$$\hat{\rho}_x(h) = \hat{\gamma}_x(h)/\hat{\gamma}_x(0).$$

The CCF - *Cross Correlation Function* is used to study the correlation when relating to time series $x[t+h]$ and $y[t]$, $h = 1, 2, \ldots, m$, and is obtained by scaling the *cross covariance function*, based on the following algorithm:

$$\bar{x} = \frac{1}{n}\sum_{t=0}^{n-1} x[t], \bar{y} = \frac{1}{n}\sum_{t=0}^{n-1} y[t],$$

$$\hat{\gamma}_{xy}(h) = \frac{1}{n}\sum_{t=0}^{n-h-1} (x[t+h] - \bar{x})(y[t] - \bar{y}),$$

$$\hat{\rho}_{xy}(h) = \frac{\hat{\rho}_{xy}(h)}{\sqrt{\hat{\rho}_x(0)\hat{\rho}_y(0)}}.$$

In practice, the function $\hat{\rho}_{xy}(h)$ is computed for a number of positive and negative values, $h = 0, 1, -1, 2, -2, \ldots$ up to some limit and the results are plotted as a function of the lag h.

Another measure related to a time series is PACF - *the Partial AutoCorrelation Function*, being useful in the identification of autoregressive models for time series.

In order to estimate PACF, let us denote by $y[t]$ the time series $x[t] - \bar{x}$ (the deviation from the sample mean). For a given lag value h, by minimizing the expression

$$MSE = \sum_{t=0}^{n-h-1} \left(y[t+h] - \sum_{k=1}^{h} a[k] y[t+h-k] \right)$$

over the possible values of the coefficients $a[1], a[2], \ldots, a[h]$, the partial autocorrelation coefficient is obtained as the estimation $\hat{a}[h]$.

There are two main types of correlation: *autoregressive*, and *moving average*.

The models denoted by $AR(p)$, respective $MA(q)$, are given by:

$$AR(p) : x[t] = a[1]x[t-1] + a[2]x[t-2] + \ldots + a[p]x[t-p] + \epsilon[t],$$

$$MA(q) : x[t] = \epsilon[t] - b[1]\epsilon[t-1] - b[2]\epsilon[t-2] - \ldots - b[q]\epsilon[t-q].$$

Combining the two models, the $ARMA(p, q)$ model is obtained:

$$x[t] = \sum_{i=1}^{p} a[i]x[t-i] + \epsilon[t] - \sum_{j=1}^{q} b[j]\epsilon[t-j].$$

Testing for stationarity is based on *unit root* approach. The following methods are mostly used: ADF - Dickey & Fuller (1979), PP- Phillips-Perron (1988), KPSS - Kwiatkowski et al (1992). The applicability in different fields is proved by Enders (1995), Hamilton (1994), and Tsay (2002), to mention only a few relevant references.

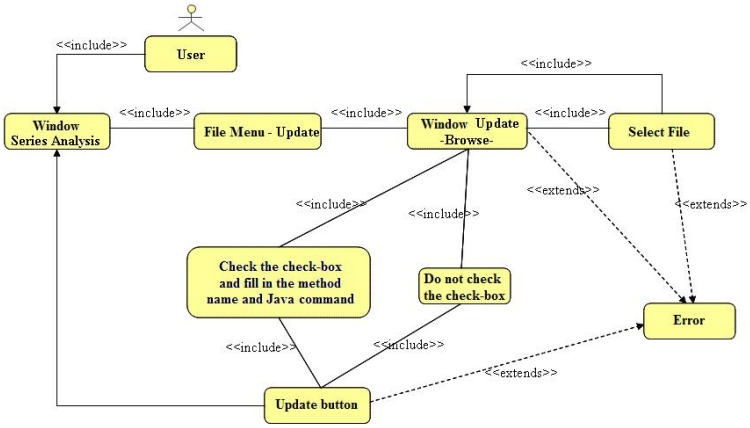

Figure 1. *SADM State Transition Diagram*

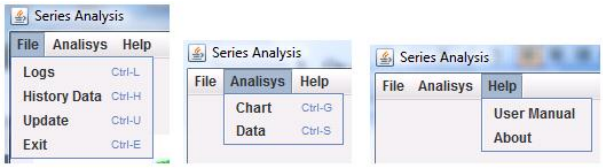

Figure 2. *SADM Front Panel*

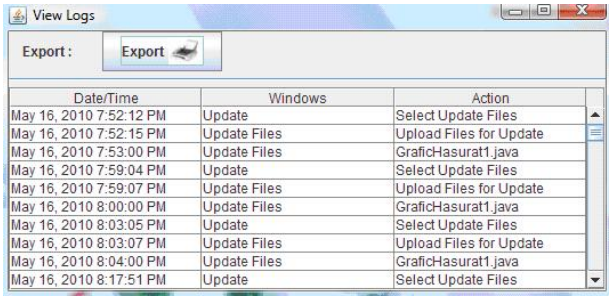

Figure 3. *Viewing the Extendability Option*

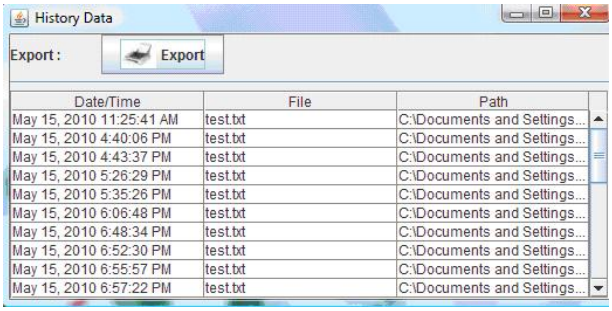

Figure 4. *Viweing the History of Time Series File*

3. **The extendability approach**

Reusability and extendability are important attributes of the software quality, as Moses J. (1994) remarked. Combining object-oriented design and agile development methodologies the software quality can be continuously improved. By reusability, the development time becomes shorter, and the usage of already existing reliable components makes possible the production of software having high reliability. At least the following artifacts can be reused: the software engine, some requirements, the source code having zero defects, the reusable data, some sections of the HELP files, some test cases, test plans, and test scripts, some user documents, and reusable human interfaces.

Extendability is the ability to incorporate new features. The extendability can be obtained by many methods: *inheritance*, *overloading*, *plug-in services*, *dynamic libraries*. New methods or services can be added or already existing methods or services can be updated when the extendability is required.

Inheritance and overloading are object oriented programming concepts. The following types of inheritance are valid: model inheritance (subtype, view, restriction, extension), variation (functional, data type, uneffecting), and software inheritance. The inheritance can be simple or multiple. This approach promotes the bottom-up design of applications.

The overloading is a software engineering process whereby multiple functions of different signature are defined with the same identifier. Both methods

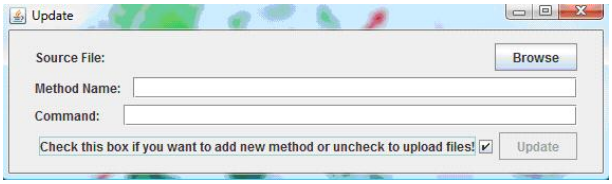

Figure 5. *SADM - The Updating Dialog*

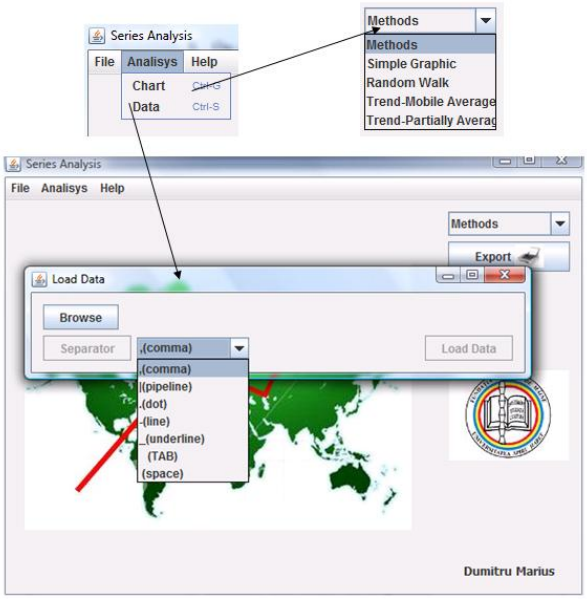

Figure 6. *SADM Analysing Data - Already Supported Methods and Options*

and operators can be viewed in multiple roles in C++ and C#, but in Java the overloading applied only to methods.

In general by plug-in components some specific capabilities or services are added to the software.

Dynamic libraries are modules that contain functions and data that can be used by other modules. Therefore, the model of the software architecture is a graph not necessary a tree.

SADM was designed in order to support updating. The state diagram is given in Fig. 1.

4. **Remarks and future developments**

The SADM architecture is the simplest possible (Fig. 2). Entire activity is registered: updating methods (Fig. 3) and time series data (Fig. 4). Updating a method asks for the method name and the specific command (Fig. 5). The updating procedure is applied for every code file to be uploaded. Fig. 6 gives a view on the analysis menu (both loading data with various separators and charting according to a selected data analysis method). A data chart for

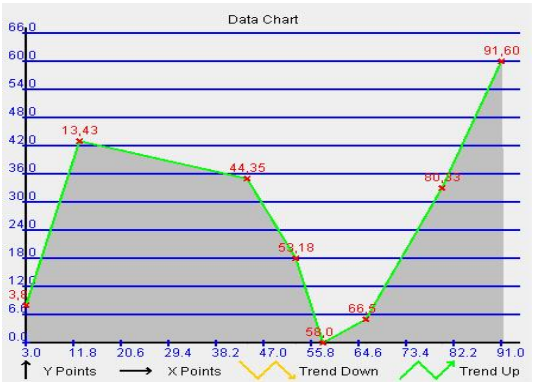

Figure 7. *SADM - A Data Chart Example*

a particular time series file is shown in Fig. 7.

In order to create extendable software the NetBeans 6.0.1 tehnology was selected. A special procedure for sending commands has been developed and the project SADM for time series analyses is the result of such extendability procedure.

SADM will be populated with new methods implementing algorithms for time series data mining. We hope that SADM will be a valuable tool not only for educational purposes but also in research.

Acknowledgement. This paper is an extended version of the presentation [1] given at ENBIS 2010 event in Antwerp. The software project SADM was implemented in Java [11] by Marius Dumitru during his master program at Spiru Haret University [6].

The authors cooperated on time series subject for many years and appreciate that actual solution will be a valuable platform for future cooperation between their institutions.

References

1. Albeanu, G., Madsen, H., Popenţiu-Vlădicescu, Fl., Dumitru, M., *An Extendable Java-based Tool for Time Series Analysis*, "ENBIS 2010", Antwerp, CDROM.
2. Albeanu, G., Madsen, H., Burtschy, B., Popenţiu-Vlădicescu, Fl. & Ghica, Manuela, *Bootstrapping Time Series with Application to Risk Management*, "R & RATA, Electronic Journal of International Group on Reliability", Vol 1(3), 84-93, 2008.
3. Albeanu, G., Serbanescu, L. & Popenţiu-Vlădicescu, Fl., *On Teaching Data Analysis and Optimisation Using Software Tools*, In Grigore Albeanu, Dorin Mircea Popovici, Marin Vlada (eds.), "Proceedings of the 2nd International Conference on Virtual Learning", Constanţa, 26-28 October, Romania, 255-260, 2007.

4. Burtschy, B., Albeanu, G., Boroş, D.N. & Popenţiu, Fl., *Improving Software Reliability Forecasting*, "Microelectronics and Reliability", Vol. 37(6), 901-907, 1997.
5. Dickey, D.A., Fuller, W.A., *Distribution of the Estimators for Autoregressive Time Series with a Unit Root*, "Journal of the American Statistical Association", 74, p. 427-431, 1979.
6. Dumitru, M., *Time Series Analysis*, "Master Thesis", *Spiru Haret* University (in Romanian), 2010.
7. Enders, W., *Applied Econometric Time Series*, "John Wiley", 1995.
8. Gheyas, I.A., Smith, L.S., *A Neural Network Approach to Time Series Forecasting*, "Proceedings of the World Congress on Engineering", July 1 - 3, 2009, London, U.K., Vol II, WCE, 2009.
9. Hamilton, J.D., *Time Series Analysis*, "Princeton University Press", 1994.
10. Hong, D.H., *A Note on Fuzzy Time-series Model*, "Fuzzy Sets and Systems", 155(2), 309-316, 2005.
11. Java Technology, http://www.sun.com/java/, 2010.
12. Kwiatkowski, D., Phillips, P., Schmidt, P., Shin, Y., *Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root*, "Journal of Econometrics", Vol 54, 159-178, 1992.
13. Madsen, H., Albeanu, G., Burtschy, B. & Popenţiu-Vlădicescu, Fl., *Addressing Time Series Modelling, Analysis and Forecasting in e-Learning Environments*, In Marin Vlada, Grigore Albeanu, Dorin Mircea Popovici (eds.), "Proceedings of the 1st International Conference on Virtual Learning", Bucharest, 37-44, 2006.
14. Moses, J., *Re-usability and Extendability in Object-oriented and Object-based design*, "Transactions on Information and Communications Technologies", 9, 427-440, 1994.
15. Phillips, P.C.B, Perron P., *Testing for a Unit Root in Time Series Regression*, "Biometrika", 75, 335-346, 1988.
16. Popenţiu - Vlădicescu, F., Burtschy, B. & Albeanu, G., *Time Series Methods for Modeling Software Quality*, "Proceedings of ESREL 2001", Italy, 1, 9-15, 2001.
17. Tsay, R. S., *Analysis of Financial Time Series*, "John Wiley & Sons", 2002.