



# The distribution of computing services for buildings design and optimal management: cloud computing based on software component standard

Benoît Delinchant, Sana Gaaloul, Pierre-Yves Gibello, Franck Verdière,  
Frédéric Wurtz

## ► To cite this version:

Benoît Delinchant, Sana Gaaloul, Pierre-Yves Gibello, Franck Verdière, Frédéric Wurtz. The distribution of computing services for buildings design and optimal management: cloud computing based on software component standard. IECON 2012, Oct 2012, Montréal, Canada. hal-00751905

**HAL Id: hal-00751905**

**<https://hal.science/hal-00751905>**

Submitted on 21 Jan 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The distribution of computing services for buildings design and optimal management: cloud computing based on software component standard

Benoît Delinchant<sup>1</sup>, Sana Gaaloul<sup>1</sup>, Pierre Yves Gibello<sup>2</sup>, Franck Verdière<sup>1</sup>, Frédéric Wurtz<sup>1</sup>

<sup>1</sup>G2elab: Grenoble Electrical Engineering

<sup>2</sup>EBM WebSourcing

benoit.delinchant@g2elab.grenoble-inp.fr

**Abstract-** The optimal building management requires developed models to predict building's systems behaviour. We propose to achieve this goal, by making available on the web, remote computation services for the composition, simulation and optimization of buildings components. A paradigm shift is necessary in order to fulfil this goal. The well suited paradigm is software components, associated with web-services technologies. We show that models can be embedded in software components and after that, made available on the web without any special effort of the user. We illustrate the use of a web-service calculation by the realization of a remote optimization without ever holding the model.

## I. INTRODUCTION

Building, with its huge primary energy consumption represents a major challenge for the coming years. So, many efforts must be done to reduce its consumption and CO<sub>2</sub> emissions by thermal insulation and optimal energy management systems using. This paper proposes to provide and describe new approaches that can be used to help buildings design refinement (simulation, optimization...) and supervision (proactive and reactive optimal management ...) improvement. This will be ensured due to the increasingly accessibility of computer networks in buildings and a major phenomenon: the cloud computing [1].

## II. SMART BUILDINGS AND OPPORTUNITIES PROVIDED BY THE "CLOUD COMPUTING"

### A. Energy networks and computer networks

The following figure (Fig 1) shows a possible link between two ubiquitous networks, that of information and that of energy. Indeed, we can imagine that all building's energy components are interconnected via two channels: energy and information technology. This is the architecture that we can call smart building: *"The use of networked technology, embedded within architecture to monitor and control elements of the architecture for exchange of information between users, systems and buildings."*, as defined by Christopher Dye, Donny Chou, Shipra Gupta, Scott Hartmann. This link may be done in reality or can be a virtual one. Indeed, we can consider that in the near future it is unrealistic to make "intelligent" all our household appliances or offices. By cons, as the same way that measurements performed on this equipment may be available on standard data servers (like OPC<sup>1</sup>); it is conceivable that models can be

also available via servers which make virtual equipments and its operating. These models can be supplied by equipment manufacturers, and made available on computational servers to provide design solutions and / or "smart" energy management.



Fig 1. The prospect of "cloud computing" in the service of the simulation and supervision of buildings

### B. The cloud computing

In this section we will outline the principle opportunities and benefits allowed when computing capabilities, simulation and optimization are available in remote web-servers:

- Sustainable development: pooling of computing resources on shared and virtualized computers which will allows energy saving.
- Dependability: services and shared data are stored in reliable and high efficient infrastructure.

<sup>1</sup> OPC: standard for data interoperability between sensors / actuators and supervision of a building's energy management system

- Scalability and easier maintenance: a shared service can be improved or maintained from distance without any need of redeployment or local effort in each post.
- Lower costs: availability of a possible use of unitarily expensive services. But the cost of these services is reduced when shared between many users (licenses computing, heavy calculations for forecasting / optimization...)
- Innovation facilities ("mashup" assembly of many services generating new uses, possible linkage with "smart grid", new actors emergence...)
- Possibility to have unlimited storage capacities and more efficient computing capabilities (at least temporarily).

### III. A DEDICATED ARCHITECTURE: FROM SOFTWARE COMPONENTS TO WEB SERVICES

#### A. Needs for paradigm changing

However, these prospects can not be achieved without a paradigm shift in applications structures that are currently used for design, simulation and monitoring. This requires going beyond the paradigm of environments, devices [2], beyond multiple software<sup>2</sup> used today in engineering teams to perform building's design, or implemented in situ to achieve its technical management [3].

That paradigm relies on non-distributed and sometimes not interoperable computing environments. For example, models are generally developed in specific software tools that, sometimes, do not offer any interoperability solution (like the initial version of COMFIE [4]). As that, it can not be used only in this tool and can not be exported or connected to other different-issued models. However, this sector is requiring more flexibility and modularity to satisfy its increasing complexity needs for a global simulation of all its components and their interactions. It is also an evolutionary area that assists, every day, to the development of new equipments and new technologies that should be integrated in control strategies and so in the global modelling.

That is why we propose to use a recent paradigm of computer architectures structure that is based on the concept of open framework, itself based on the concept of autonomous software component that are conceptually dedicated for interoperability. The component approach follows the oriented object paradigms in the history of structuring software architectures in computer science [5] (Fig 2). It allows weaker coupling and so more flexible models.

It is in this goal that was initiated the ICAR pattern as a standard for software components, which can be used in particular in the building simulation domain [6].

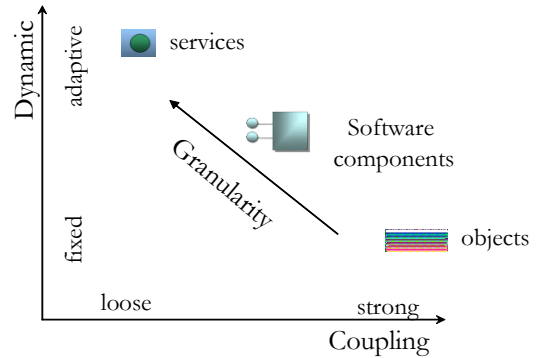


Fig 2. Modelling paradigms comparison

#### B. Advantages of the software component based approach:

Many solutions have been proposed to improve cloud services interoperability by, for example, proposing cloud computing standards like [7] [8]. We propose, in this paper, the use of a software component standard named ICAR

Software components can automatically treat and manage different issues such as communication between applications, heterogeneity of programming language from which it is generated (plug' out) or in which it can be use (plug' in), and between operating systems as it is platform independent. It also allows capitalization and making available "on shelf" complex models in the case of composed systems.

A standard of software components specifies its interfaces that define the kind and the format of information that will be exchanged by the component. This standardization approach allows easier models sharing because models will have known and normalized interfaces. Moreover a software component must also specify its:

- Packaging and encapsulation methodology: how it can be generated and how it is composed.
- Deployment and introspection mechanism: how to use the component, to interrogate it, and to know its available services.
- Offered interoperability process: via http or web services for example.

Many ICARs plug' out have been already developed to allow automatic component generation respecting the ICAR pattern. In the other side, a tool that has already implemented this standard will receive and use any ICAR component (plugged in) without any difficulties.

Thus, we can note that the ICAR standard, operating for ten years in the field of optimal design of electrical engineering systems [9], is able now to offer efficient solutions for interoperability between tools or languages such as Modelica [10], Pleiade / Comfie [6], TRNSYS, etc...by exchanging models via a software component bus.

So we can summarize the expected benefits of the software component standard in terms of:

- Capitalization / reuse:
  - Models or shortcuts available on shelf and ready to reuse.

<sup>2</sup> [http://apps1.eere.energy.gov/buildings/tools\\_directory/](http://apps1.eere.energy.gov/buildings/tools_directory/)

- Reuse of robust codes that are already tested and with results are guarantee.
- **Composition:**
  - Systems construction by assembling many components that are already available,
  - Improved modularity, more abstract and functional development, dynamic extensions by composition...
  - Portable and distributable systems as the software component is an executable autonomous code.

An important feature of these components is their autonomy. So, they can be exchanged between partners, or simply made available for download on dedicated sites for capitalization / re-use of models such as French project DIMOCODE (<http://www.dimocode.org>).

The next step is to make the computing capacity of a model, encapsulated in a component, directly usable, remotely, on a web-server.

### C. Made available on Internet through Web Services

#### 1. Some recommendations

The solution we have implemented respect the following technical recommendations:

- A technology based on a common Internet Protocol: HTTP, REST infrastructure (*Representational State Transfer*)
- Deployable in a technical foundation for common use: Java Web Application (*Apache Tomcat application server*)
- Maximum neutrality ahead protocols, technologies and languages (*little normative approach*): JSON encoding.

Components specified flowing recommendations are easy to develop, implement and assemble, and induced no limitation for future innovation development.

#### 2. Illustrating Example

We consider a simple model of wall heat conduction (Fig 3). Fourier's law applied to the equation (1) gives the heat flux density expression (2) and thus the total flow (3).

We will simply describe the model using the final equation (4) giving the difference of temperature as a function of the heat flux and the thermal resistance of the wall.

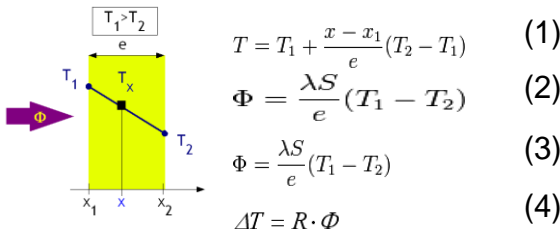


Fig 3. Simple case of heat conduction in a wall

This model can be described simply in the tool CADES ([www.cades-solutions.com](http://www.cades-solutions.com)) [9] which then generates a software component ICAR (*Interface for Component Architecture*) [11].

The generated ICAR component is also compatible WAR (*Web Application ARchive*). That is why, it is sufficient to download it via FTP (*File Transfer Protocol*) in the folder of a Tomcat format application server (*tomcat. apache.org*) to

have a deployable calculation that is accessible via web-services (Fig 4).

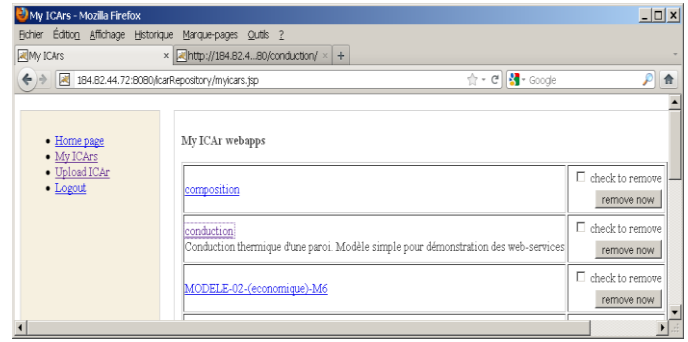


Fig 4. Web interface to make available a web-service model

We also developed a web interface to download the component on the server, accessible by login, which allows its services managing (Fig 5):

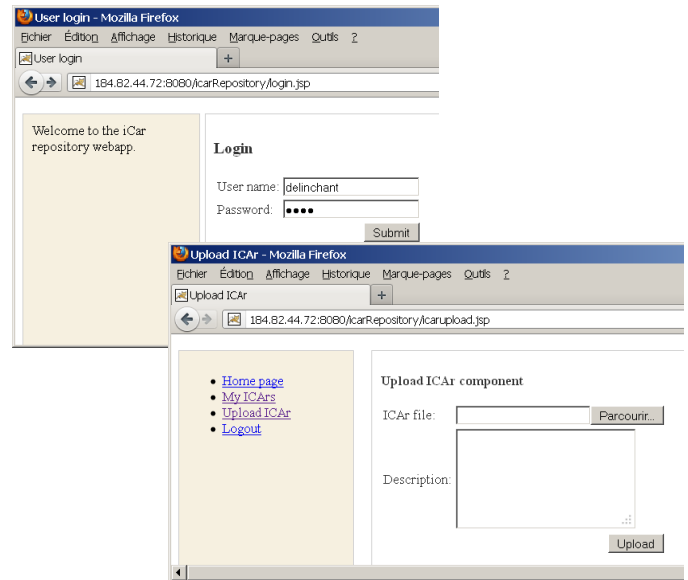


Fig 5. Web interface to manage the model web-services

Once loaded, the list of different services offered by the component and its ports details can be displayed by a simple http query (Fig 6).

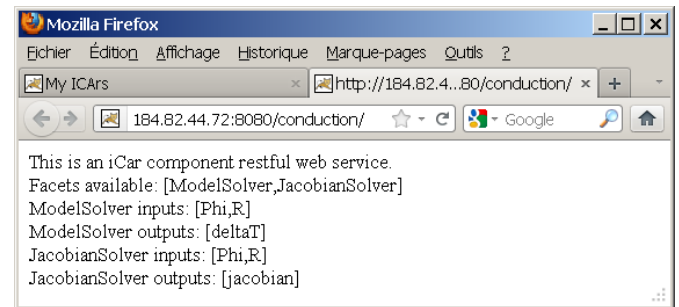


Fig 6. List of services and ports of the "conduction" web-service

The query is formulated in HTTP on an URL that represents a resource. The query is composed of parameters (*QUERY\_STRING part*) that must be interpreted by the resource. This can be simply performed in the web browser, but especially in any language (Python, Java, Net ...) and on very easy way.

For example, to know available services in the component, it is enough to simply run the following query:

Query:

`http://184.82.44.72:8080/conduction/facets`

Reply: [ModelSolver, JacobianSolver]

The component and its facets provide also some metadata description such as the list of their input and output ports. For example, input ports can be obtained by typing:

Query:

`http://184.82.44.72:8080/conduction/facets/ModelSolver/inputnames`

Reply: [Phi, R]

It is now possible to invoke this facet to perform a calculation like (calculation of deltaT, for phi= 2 and R = 3):

Query:

`http://.../conduction/facets/ModelSolver/resolve?{R:3.0, Phi: 2.0}`

Reply: {deltaT: 6}

#### IV. CASE STUDY: DESIGN AND OPTIMAL MANAGEMENT OF AN ELECTRICAL HEATER COUPLED TO A THERMAL ENVELOPE

##### A. The use case description

The application case illustrating our purpose consists on the sizing of a heating system associated to an optimum revival temperature that allows consumed energy minimizing with respect of user comfort and its set points (temperature settings). To do this, two models are available:

- A building envelope model that calculates the internal temperature as a function of internal (heating power) and external (external temperature) sources.
- An electrically controlled heater, injecting a heating power as a function of a set point and the current temperature.

CADES is used to describe the model compound, this form of algebraic equations and algorithms:

##### B. The CADES framework

CADES (*Component Architecture for the Design of Engineering System*), is a framework dedicated to system design and optimization. It is developed around the ICAR standard (Fig 7).

This framework allows the ICAR generation using Cades generator through a SML (System modelling language)

language description or by composing two or more already generated components as it is a composition environment.

An ICAR component generated from Cades framework or another tool due to automatic developed plug' out, can be imported in the Cades framework to benefit from its services like, computation, optimisation, parameters sensitivities study, composition...

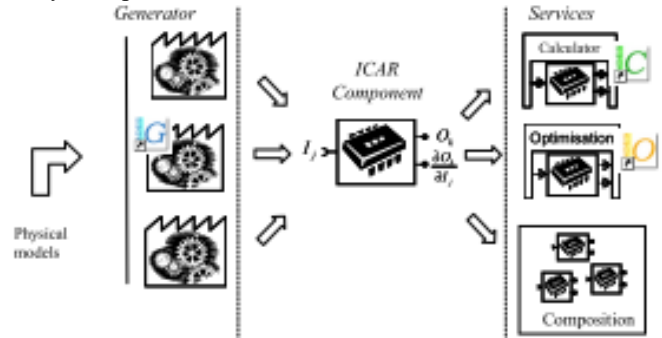


Fig 7. The CADES framework

##### C. The system's ICAR generation and virtual optimization in CADES via web servers

Cades was used to describe the composed model of the global system envelop + heater through algebraic equations (Fig 8) and algorithms form.

The software component is then automatically generated (model analysis, definition of computing graph incidence, Java code generation, compilation, packaging).

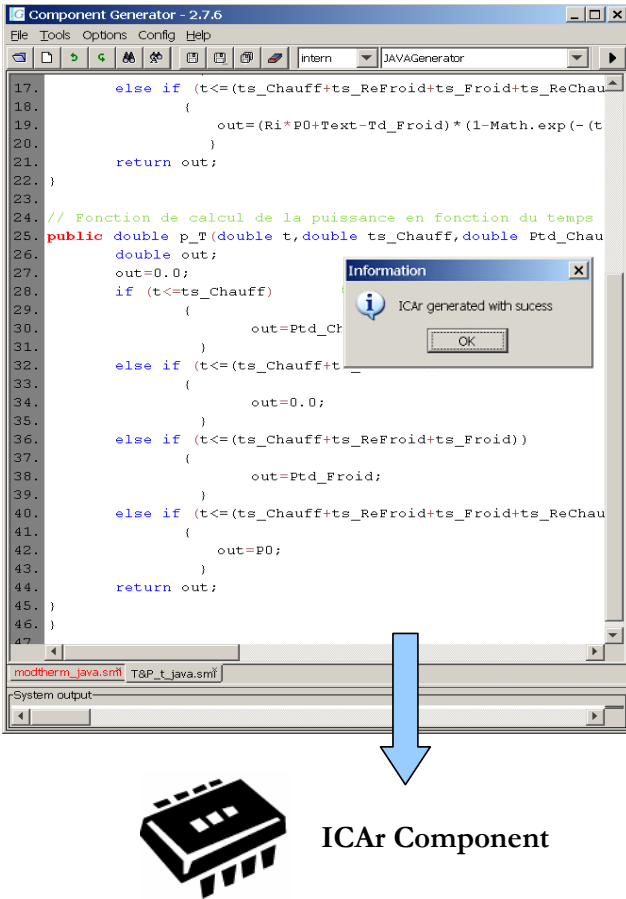


Fig 8. ICAr component generation from CADES environment

After that, the corresponding file is deposited on an application server via a web browser. It becomes automatically available and ready for use as a web service. This web service can be then connected to an optimization tool.

The realized optimization can, for a set of weather data and envelope characteristics (R and C parameters), found an optimal sizing of the heater's rating power heating and its associated optimal control that satisfy comfort instructions.

The following figure (Fig 9) illustrates this optimization in which the optimized system is virtually present and connected to an optimization algorithm.

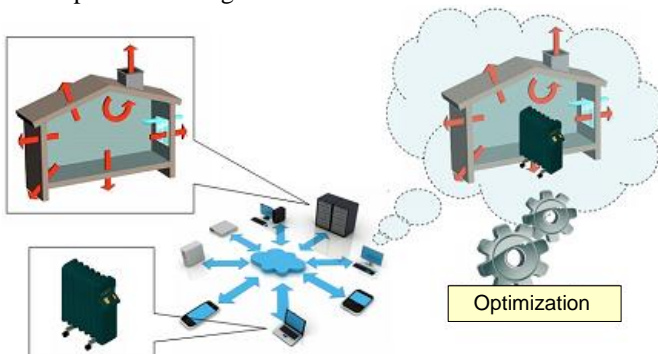


Fig 9. An example of an optimization using distributed components that are composed / orchestrated in the "cloud"

The figure at the end of the paper (Fig. 11) shows that the optimization is performed in 10 iterations and 3.7 seconds. It is a fast optimization thanks to the availability of the Jacobian of the model.

Optimization results are used to determine the heating power ( $P_0 = 100W$ ), the optimal revival temperature ( $T_{dfroid} = 2^\circ C$ ) which allows day energy minimizing ( $E_{total} = 647J$ ).

## V. CONCLUSION

In conclusion, we propose a new vision for the design and optimal management of buildings. In this view, models are created and made available on application servers through the web and then used during the phases of design, but also during the forecast based optimal management embedded in buildings.

## REFERENCES

- [1] L. Wang and G. von Laszewski, "Scientific Cloud Computing: Early Definition and Experience", *High Performance Computing and Communications*, 2008.
- [2] S. Van der Burg, E. Dolstra, M. de Jonge, E. Visser, "Software Deployment in a Dynamic Cloud: From Device to Service Orientation in a Hospital Environment", *Software Engineering Challenges of Cloud Computing*, CLOUD'09, May 2009.
- [3] H.T. Mustafa and P.K. Bansal, "Building Management Systems: Beyond Electronics", *THE OFFICIAL JOURNAL OF AIRAH*, p.p 22-27.
- [4] B. Peuportier, I. Blanc-Sommereux, "Simulation tool with its expert interface for the thermal design of multizone buildings", *International Journal of Sustainable Energy*, vol 8, issue 2, p. 109-120, 1990.
- [5] C. Szyperski, *Beyond object-oriented programming*, Boston, Addison-Wesley Professional, 2ème edition 2002.
- [6] S.Gaaloul, B. Delinchant, F. Wurtz, F. Verdier, "Software components for dynamic building simulation", *Building simulation conference*, Sidney, Australia, 2011.
- [7] IEEE Cloud Computing Standards Committee, "IEEE Standards Association (IEEE-SA) Sponsor Baseline Policies and Procedures for Standards Development", August 2011.
- [8] A. Sampaio, N. Mendonça, "Uni4Cloud: an approach based on open standards for deployment and management of multi-cloud applications", *SECCLOUD '11 Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing*, New York, USA, 2011.
- [9] B. Delinchant, F. Wurtz, D. Magot, L. Gerbaud "A component-based framework for the composition of simulation software modelling electrical systems", *Journal of Simulation, Society for Modelling and Simulation International*, Special Issue: Component-Based Modeling and Simulation. Jul 2004; vol. 80: pp 347 – 356
- [10] F. Verdier, A. Rezgui, S. Gaaloul, B. Delinchant, F. Wurtz, X. Brunotte, "Modelica models translation into Java components for optimization and DAE solving using automatic differentiation", *USKM*, 2012.
- [11] B. Delinchant, D. Duret, L. Estrabaut, L. Gerbaud, H. H. Nguyen Huu, B. Du Peloux, H.L. Rakotoarison, F. Verdier, S. Bergeon, F. Wurtz F., "An Optimizer using the Software Component Paradigm for the Optimization of Engineering Systems", *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 26 No. 2, pp. 368-379, 2007.

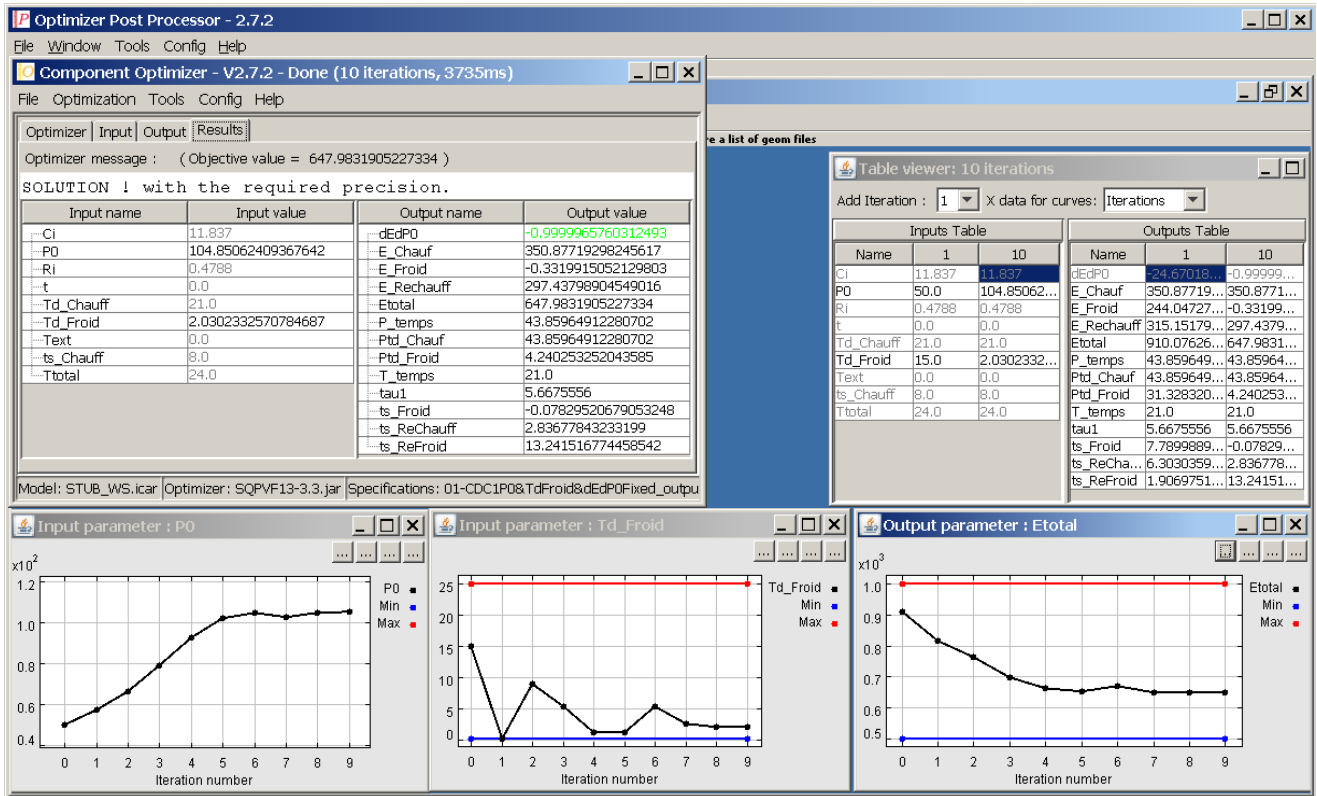


Fig 10. Results of the optimization of the web-service model