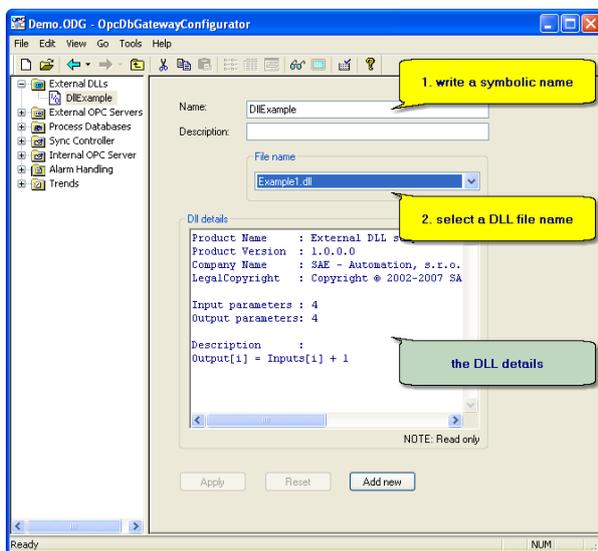


## Customization of the runtime and configuration functionality

**Configuration application of the OpcDbGateway** enables integration of applications for handling data from many different data sources by **configuring, which is mostly less labourious process than application programming.**

Using of configuring with specialised configuring application in OPC servers, is often used and very useful feature provisioning that it is not necessary to make changes in the source code of the communication driver for every change in the set of communicated data. The information about this data set (address space of the OPC server) is put to the configuration file from which the communication driver can read it and adapt the functionality of the communication driver accordingly. The same driver can be then used in many different applications with application specific configurations.

This principle was enhanced within OpcDbGateway **from the configuring of the communicated data sets also to the configuring of data processing, saving to the process database files and providing processed data for visualisation or other external applications.**

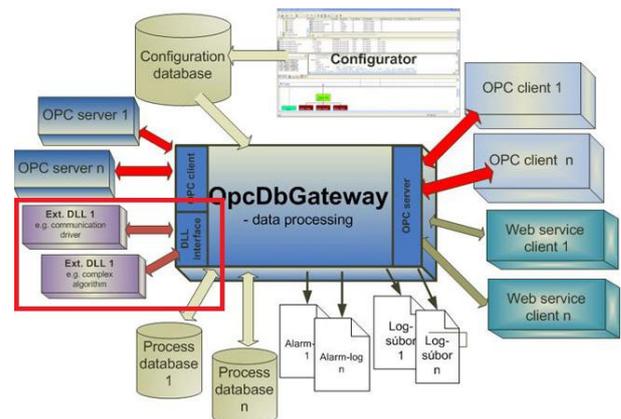


**Fig. 1:** New external DLL connection in OpcDbGateway.

Despite of **high productivity of the applications configuring**, it is very useful sometimes **to have possibility to program some parts of the application or to create new modules repeatedly usable in configured applications.**

New modules can be created as **dynamic linked libraries (DLL) used by both applications – by the runtime of the OpcDbGateway and by configuration application.** Dll-functions enabling to read information about Dll as name, vendorinfo, the input and output parameters are used within configurator. Within the runtime, the main function of the enhancement DLL is called to fulfil the core functionality of the enhancement DLL.

OpcDbGateway enables, first of all, processing of data from external OPC server and client applications and databases (Fig.2). The enhancement DLL provides for OpcDbGateway substantial enhancement of accessible data sources as well as the enhancement of the algorithms classes which can be run in the OpcDbGateway runtime.



**Fig.2:** Processing of data from external data sources in OpcDbGateway.

**Various communication drivers can be implemented as an enhancement DLL.** It avoids the necessity to have external OPC server for every device from which we want to process data in OpcDbGateway. This way, for example the DDE communication, SMS and e-mail messaging, calling of web services can be implemented. If we create own communication driver as the enhancement DLL, the OpcDbGateway is becoming OPC server/client for such communication driver. In this meaning, OpcDbGateway can be perceived as universal OPC client/server application.

**Various complex algorithms as for example digital filters, regulators and simulation models of the dynamic systems,** which must be completely fulfilled within one period of the internal synchro-

nous controller of the OpcDbGateway, can be also implemented as the enhancement DLL.

The OpcDbGateway usage scenarios can be this way completed with: (Fig. 3):

- a bridge between functionality provided by 2 or more DLL and DLL,
- a bridge between DLL and OPC server,
- a bridge between DLL and OPC client
- a bridge between DLL and web client.

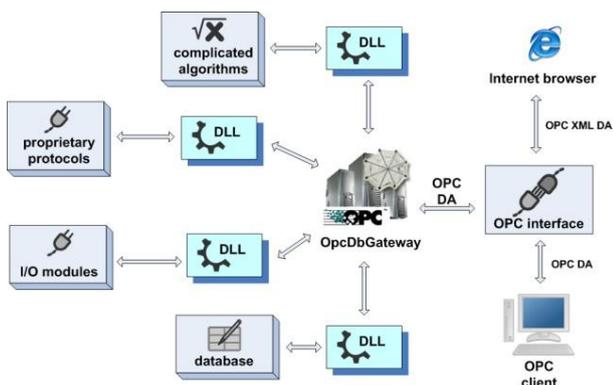


Fig. 3: External DLL, the new possibility how to enhance OpcDbGateway functionality.

Essential advantages resulting from integration of the external DLL to the OpcDbGateway application can be summarized as: (1) an enhancement of OpcDbGateway functionality (2) a possibility for customers to build their own product including all functionality of the OpcDbGateway runtime as well as configuration features offered by the OpcDbGateway configuration application.

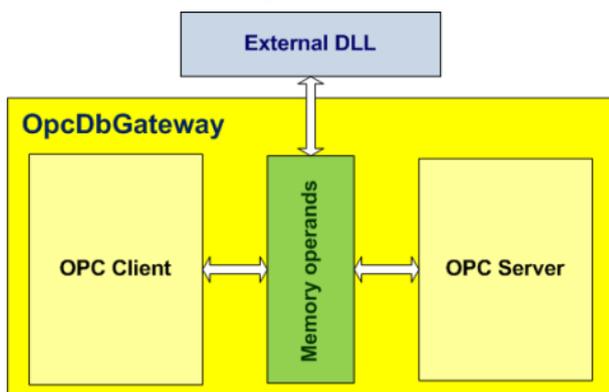


Fig. 4 Cooperation of the different datasources using memory operands.

An interconnecting of the enhancement DLL with another functionality of the OpcDbGateway is based on using of shared memory area with variables of the type VARIANT called memory operands. The configuration application enables mapping of the internal variables of the internal OPC server, internal OPC client to the named

memory operands with defined position in the shared memory area (Fig. 4). The enhancement DLL has an interface with the memory operands area over input and output parameters of the main function used by the enhancement DLL. The functionality of the OpcDbGateway configurable by configuration application (Fig. 1) is based just on using of memory operands shared by all configurable parts of the OpcDbGateway runtime.

External DLL uses continuous array of memory operands as input and output parameters of the main function of the enhancement DLL DoProcessIO:

```
DllExport bool WINAPI DoProcessIO(
    const CComVariant* lpInputs, WORD
    wInCnt,
    CComVariant* lpOutputs, WORD wOutCnt
);
```

The function has next parameters:

- lpInputs[in] - Pointer to the buffer that includes input values.
- wInCnt[in] - Size of the buffer pointed to by the lpInputs parameter, in CComVariant lpOutputs
- [out]Pointer - to the buffer that receives output values.
- wOutCnt[in] - Size of the buffer pointed to by the lpOutputs parameter, in CComVariant

If the function succeeds, the return value is non-zero. If the function fails, the return value is zero.

The function will be called from OpcDbGateway runtime within a configurable Function Block using the configurable command CALL DLL with next functionality:

```
CALL DLL(Output, Input1, Input2)
```

**Result:** Output - Memory operand. If the function succeeds, the return value is non-zero. If the function fails, the return value is zero.

**Parameters:**

- Input1 - the external DLL name
- Input2 - memory operand - the first item in the array of memory operands

The array of memory operands consists of two parts: (1) defined number of input memory operands (2) defined number of output memory operands

It means that the address of the first output memory operand is the address of Input2 plus a number of input memory operands. The content of input memory operands is passed as input parameter (lpInputs) to the external DLL function Do-

ProcessIO. In this function all input data are processed and stored to output parameter (*lpOutputs*). Afterwards, the function is finished and the output parameter values copied to output memory operands.

The complete MS Visual Studio example project for implementation of the enhancement DLL is installed to the directory `C:\Program Files\OpcDbGateway\Examples\ExternalDlls\Example1`. It can be used, as template to create a customer's DLL.

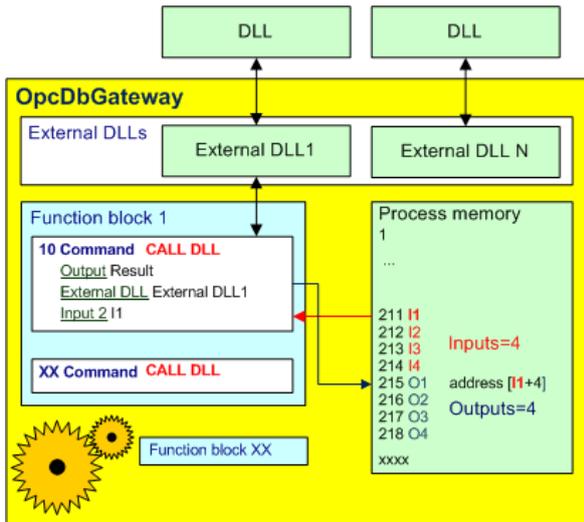


Fig. 5 Calling of the function *DoProcessIO* using configurable command CALL DLL.

The configurable calling of the function *DoProcessIO(...)* within Function block has a few causalities:

- CALL DLL command can be called within one function block one or more times.
- If we use one input memory operand as counter of the statuses of the status automat implemented in the function *DoProcessIO(...)* in subsequent calling of the same function *DoProcessIO(...)* of the same enhancement DLL will be provided different functionality. This is often used by implementation of various communication drivers.
- If CALL DLL command is placed to the function block MAIN then the *DoProcessIO(...)* will be called cyclically.
- If the CALL DLL command is placed to the function blocks START, RESTART, STOP nonrecurring fulfillment of the function is provided.
- It can be called also conditionally using different configurable triggers.
- The OpcDbGateway runtime runs as Windows service without user interface. Because of this, the enhancement DLL must also not implement user interface directly. It can only provide communication with other applications.

A good example of that is the DDE communication with MS Excel. Example of such enhancement DLL can be downloaded from: [www.saeautom.sk/download/EnhDLL\\_DDE\\_xls.zip](http://www.saeautom.sk/download/EnhDLL_DDE_xls.zip). The functionality of the enhancement DLL is described in the window „Dll details“ of the Figure 6.



Fig. 7 DDE communication between OpcDbGateway and Excel-sheet.

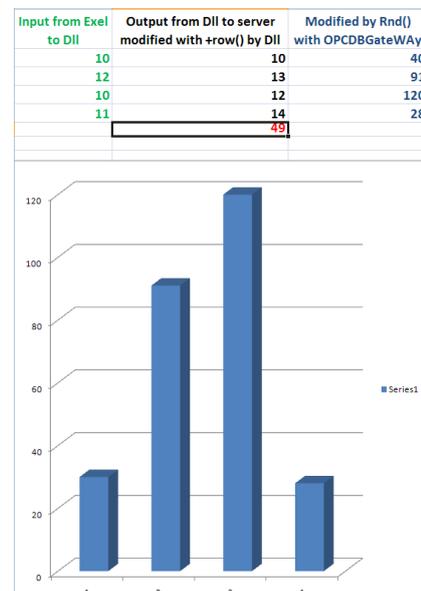


Fig. 7 Visualisation of the example application of the DDE communication in the Excel-sheet.

Other details to the enhancement dll can be found in [www.saeautom.sk/download/OpcDbGateway.pdf](http://www.saeautom.sk/download/OpcDbGateway.pdf).