For client and server to communicate by exchanging messages there has to be a set of rules both sides adhere to. These rules are called protocols. Communication models are used to hide the details of these protocols and to conceal the communication itself from the communicating processes.

At the rate which both the software and the hardware is evolving, there is pressure on the used communication protocol, too. It should be able to handle new demands without closing older clients outside the services. A flexible communication protocol makes programming clients and servers easier and allows changes in the structure of the messages without demanding major changes in the applications, if any.

In this study, three communication models are examined: Remote Procedure Call (RPC), Remote Object Invocation and Message-oriented Communication. Different text and binary protocols based on these models are presented (XML-SOAP, XML-RPC, CORBA and Java RMI) and their advantages and disadvantages in the context of flexibility is examined.

The flexibility in the client and the server applications is also closely related. Different ways to design and implement client-server applications to be flexible and to be able to prepare for changes or evolution in these communication protocols are also studied.

Keywords: flexibility, message-oriented communication, XML, SOAP, XML-RPC