Question:
Can LabVIEW communicate with Agilent instruments without the use of instrument drivers?

Answer:
Yes, though it is somewhat more difficult than using Agilent VEE's direct I/O object. LabVIEW was designed to communicate with instruments at the VISA level. As long as the instrument has a LabVIEW driver (or a plug and play driver), programming is rather straight-forward. However, when there are no drivers for the instrument, as is the case with many older instruments, programming becomes a little more difficult.
The front panel has three user-input windows and three output displays. The user inputs are as follows:

1. The first input is labeled GPIB Address. In it you must input the address of the instrument on the GPIB bus. 
2. The next window is labeled "Characters to Write". This is where you put the string you want to send to the instrument (for example: *IDN?). To enable the write function it is necessary to click on the switch located directly under the Write label. To see the "visual code" behind this program press "control e". The first block diagram that appears is the frame that allows LabVIEW to write to the instrument. The heart of this frame is the GPIB write VI. A screen of the frame is shown here:

Note that the "Mode" is set to zero. On some older instruments you may have to change the mode setting in order to achieve proper handshake. In the top center of the sequence frame (the frame that looks like a piece of film, sprocket holes and all), you will find the means to access the read function. Just click on the right arrow and the second screen will appear. This frame enables LabVIEW to read the contents of the instrument's output buffer. The heart of this frame is the GPIB read VI. The following is a screen of the read frame.
Select the VISA address of the device, choose Read or Write or both, type in the characters to be written, and run the VI. If you choose both Read and Write, the VI will write to the device first, then read from the device. If you are using a Hewlett-Packard HP-IB interface you can find the VISA Name by running the I/O Configuration utility.

VISA_Name: Instrument_Address
Example: GPIBO:09

Characters to Write

Characters Read

# bytes to read
1000
This program accepts similar inputs. However, instead of the GPIB address you will have to enter the VISA name and instrument address separated by a double colon. To find the VISA name, run the I/O config program. You will find the VISA name in the window that shows the configured interfaces.

In this screen you can see that the GPIB write VI has been replaced with a VISA write VI, and the status report has been deleted.

The read frame can be seen by clicking on the right arrow of the sequence frame just as it was done in the previous program. The screen is shown on the next page.
This screen shows the read function. Once again, the GPIB read VI has been replaced with the VISA read VI.

Summary:

The GPIB and VISA VIs enable Labview to write strings to an instrument and read the results of a query. GPIB VIs can be used with NI-GPIB cards and VISA VIs can be used with Agilent GPIB cards.

Once you have the ability to send and receive strings, you can access the instrument's remote programming capabilities. From here on, it is a question of incorporating the GPIB or VISA I/O functions in the LabVIEW instrument control program and sending the appropriate strings to the instrument.

P.S.

As mentioned above, the VEE implementation of Direct I/O is somewhat less cumbersome. To view the block diagram of the VEE 5.0 implementation of "smplsicl" see the following image. Note that the entire block diagram can be viewed at once, making it easier to understand its flow.
The front panel can be arranged as desired. This screen is an example of the front panel.