Summary

In Delphi it is possible to create your own components. Therefore, it is also possible to create components that act as conveyor belts. When many conveyor belts act together, it is called a network of conveyors. Packages start at the begin of a conveyor network and run through to the end of a conveyor network. In order to explain the full behavior of a conveyor network and the time it takes for a package to run from point A to point B, three types of conveyors must be defined. Straight conveyors, conveyors that are split into two conveyors, and conveyors where two conveyors meet and form one conveyor. These conveyor types are represented in Figure 1.

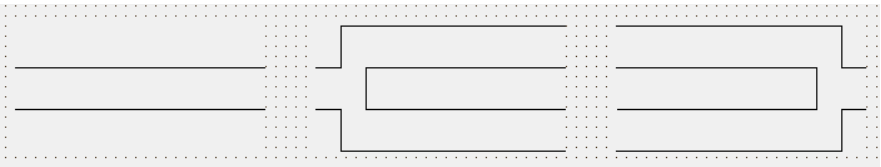

Figure 1 - Representation of a straight conveyor (left) a conveyor splitter (middle) and a conveyor meeting point (right)

For each of these conveyor types a component is created. The conveyor types need each other's information in order to determine the time it takes for a package to run through the conveyor. Therefore, the conveyor components must be able to be connected to each other. A change in the value of a property of a conveyor component may also cause a change in the value of a property of the components connected to this component. Therefore, the conveyor components need to notify and call the procedures of the components connected to them whenever a value of a property is adjusted.

As it is useful to know the position of a package in the component a property for this is added to the components as well. Due to this property it is very simple in a Tomas Delphi application to visualize the position of a package in the conveyor network.

As the three conveyor components have a lot in common, they are all three derived from the same base component. This component owns most of the characteristics of the conveyor components. The code for the conveyor splitter and the conveyor meeting point is more complicated as the package can take two routes through the components, instead of one. Together with the code of the base component, there are four units that describe the conveyor components. All units are added to the same package and the package is installed so that the conveyor components can be used.

The properties of the conveyor component can be seen in the Object Inspector. The Object Inspector and the shape of a conveyor splitter component are shown in Figure 2. Properties like 'Acceleration', 'Deceleration', 'ConveyorSpeed' etc. are visible. Also the properties 'Input' and 'Output' are shown. Using these properties, other conveyor components can be selected that can be connected to this component. The graphical representation of the package can be changed by the property 'Movingnumber'. This property can take the values 0 to 10. When the value is 0, the package is at the start of the conveyor component, and when the value is 10, the end of the conveyor component has

been reached. The position of the package (Movingnumber=4) is denoted by the black rectangle in the figure.

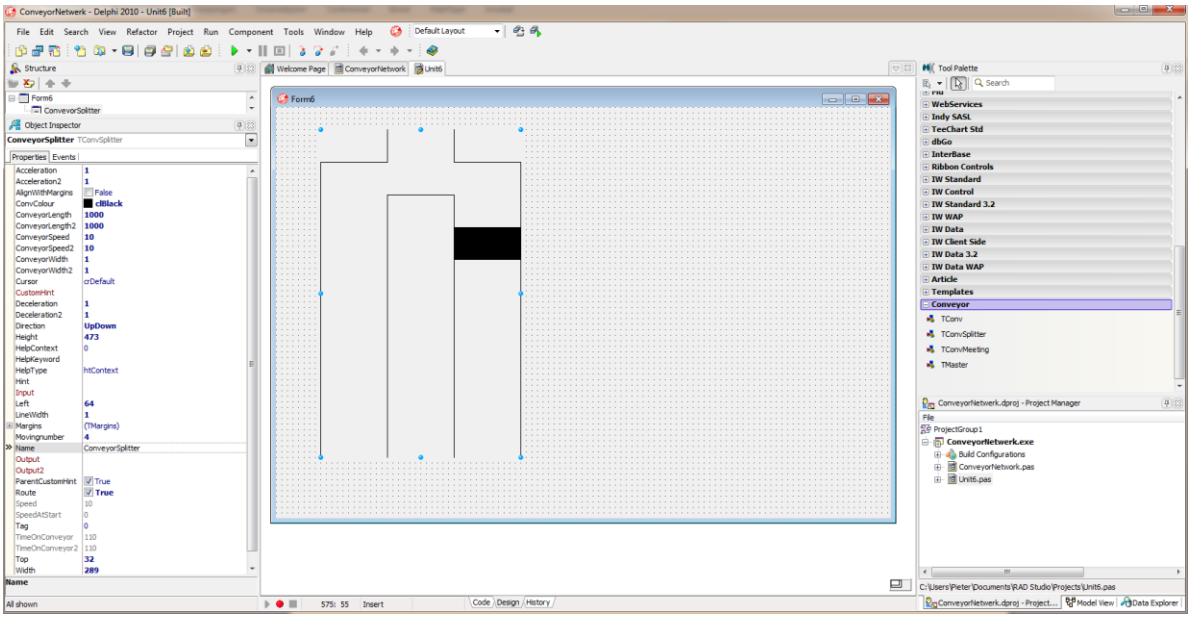

Figure 2 - Conveyor splitter on a form with opened Object Inspector

In Figure 2, also the palette page 'Conveyor' is visible in which the conveyor components are installed. When they (TConvSplitter, TConvMeeting and TConv) are selected they can be dragged on a form.

Now, the conveyor components can be used to build a large conveyor network. The components are connected to each other and use values of each other's properties. With Tomas Delphi such a networks is very easy to build as the conveyor components do most of the work. In the network a package can follow multiple routes. The time it takes for a package to run though the network can be retrieved. Figure 3 shows an example of a conveyor network with multiple routes.

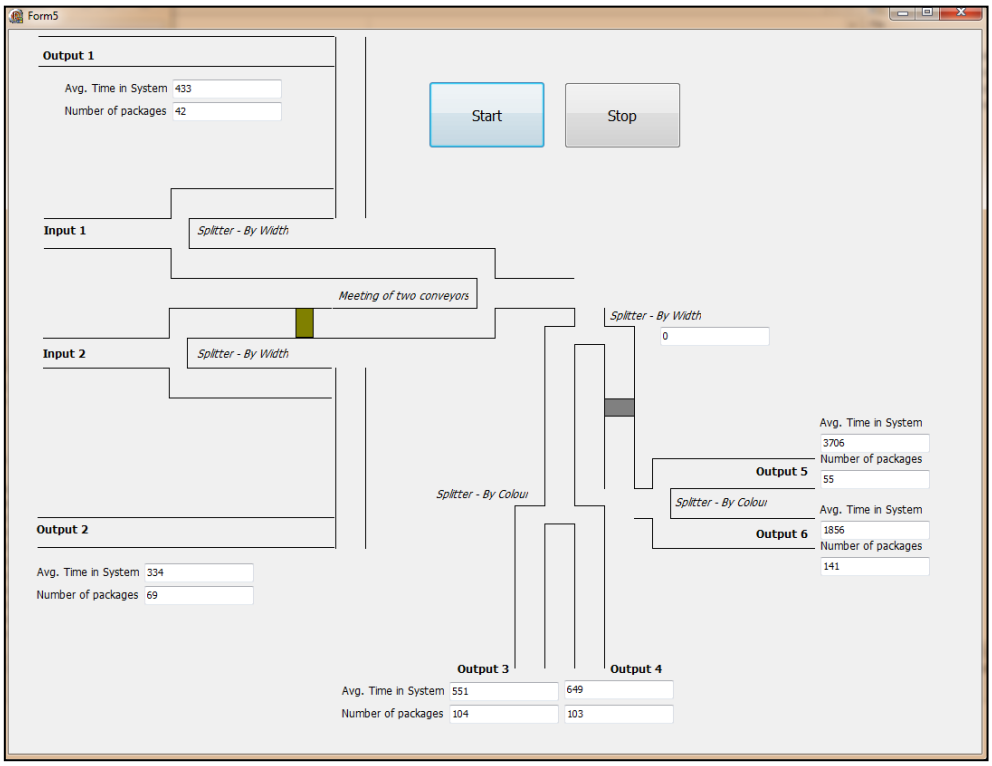

Figure 3 - Application of a conveyor network