Multi-modules application with consolidated build

This use case address the situation where a monolithic project needs to be modularized with the ability to generate a consolidated jar that mimic the current situation.

TODO list:

- Set the assembly in a profile that is activated when the release is performed
- Handle missing package.html file in source archive (javadoc generation on the fly)

Download a sample of this use case

Assuming that the project *Foo* has been split into 3 modules.

```
.
|-- pom.xml
|-- module-a
    `-- pom.xml
|-- module-b
    `-- pom.xml
|-- module-c
    `-- pom.xml
|-- foo-all
    `-- pom.xml
```

Each of the three modules have a *jar* packaging. The build generates the main build as well as the sources using the source plugin. The *foo-all* pr oject generates the consolidated build of the 3 modules (jar, sources and javadoc).

Main pom sample:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.foo</groupId>
    <artifactId>foo-application</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>pom</packaging>
    <name>Foo application</name>
    <description>
        A sample application with 3 modules with the ability to generate a
consolidated build.
    </description>

    <dependencies>
        <!-- Common dependencies of the project goes here -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <inherited>true</inherited>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-source-plugin</artifactId>
                <executions>
                    <execution>
                        <id>attach-sources</id>
                        <goals>
                            <goal>jar</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>

    <modules>
        <module>module-a</module>
        <module>module-b</module>
        <module>module-c</module>
        <module>foo-all</module>
    </modules>
</project>
```

The 3 modules have their own configuration and dependencies. Each of them extends the main pom of the application (see above). For instance, a minimal pom for module-a would look like:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.foo</groupId>
        <artifactId>foo-application</artifactId>
        <version>1.0-SNAPSHOT</version>
        <relativePath>../pom.xml</relativePath>
    </parent>
    <artifactId>module-a</artifactId>
    <packaging>jar</packaging>
    <name>The module A</name>
</project>
```

The foo-all is responsible of building the main application. It has a dependencies on the 3 modules (both the main build and the source archive). The consolidated build and source archive are generated by the assembly plugin. The Javadoc is generated on the fly by expanding the source archives.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.foo</groupId>
        <artifactId>foo-application</artifactId>
        <version>1.0-SNAPSHOT</version>
        <relativePath>../pom.xml</relativePath>
    </parent>
    <artifactId>foo-all</artifactId>
    <packaging>pom</packaging>
    <name>The consolidated module for application foo</name>

    <dependencies>
        <dependency>
            <groupId>com.foo</groupId>
            <artifactId>module-a</artifactId>
            <version>${project.version}</version>
        </dependency>
        <dependency>
            <groupId>com.foo</groupId>
            <artifactId>module-a</artifactId>
            <version>${project.version}</version>
            <classifier>sources</classifier>
        </dependency>

        <dependency>
            <groupId>com.foo</groupId>
```

```
            <artifactId>module-b</artifactId>
            <version>${project.version}</version>
        </dependency>
        <dependency>
            <groupId>com.foo</groupId>
            <artifactId>module-b</artifactId>
            <version>${project.version}</version>
            <classifier>sources</classifier>
        </dependency>

        <dependency>
            <groupId>com.foo</groupId>
            <artifactId>module-c</artifactId>
            <version>${project.version}</version>
        </dependency>
        <dependency>
            <groupId>com.foo</groupId>
            <artifactId>module-c</artifactId>
            <version>${project.version}</version>
            <classifier>sources</classifier>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-assembly-plugin</artifactId>
                <version>2.2-SNAPSHOT</version>
                <configuration>
                    <descriptors>

<descriptor>src/assemble/all-binary.xml</descriptor>

<descriptor>src/assemble/all-sources.xml</descriptor>
                    </descriptors>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-dependency-plugin</artifactId>
                <executions>
                    <execution>
                        <id>unpack-dependencies</id>
                        <phase>generate-resources</phase>
                        <goals>
                            <goal>unpack-dependencies</goal>
                        </goals>
                        <configuration>

<includeClassifiers>sources</includeClassifiers>
                            <excludeTransitive>true</excludeTransitive>

<outputDirectory>${project.build.directory}/javadoc-src</outputDirectory>
                        </configuration>
```

```
                    </execution>
                </executions>
            </plugin>
            <plugin>
                <artifactId>maven-antrun-plugin</artifactId>
                <executions>
                    <!-- generate the javadoc -->
                    <execution>
                        <id>generate-all-javadoc</id>
                        <phase>compile</phase>
                        <configuration>
                            <tasks>
                                <javadoc packagenames="com.foo.*"
                                         maxmemory="256m"

sourcepath="${project.build.directory}/javadoc-src"
                                         defaultexcludes="yes"

destdir="${project.build.directory}/apidocs/"
                                         version="true"
                                         use="true"
                                         windowtitle="${project.name} API">
                                    <doctitle>
                                        <![CDATA[<h1>${project.name} API
v${project.version}</h1>]]></doctitle>
                                    <bottom>
                                        <![CDATA[<i>Copyright &#169;
${project.inceptionYear} Bar Software S.A. All Rights
Reserved.</i>]]></bottom>
                                    <classpath
refid="maven.dependency.classpath"/>
                                </javadoc>
                                <jar
destfile="${project.build.directory}/${project.build.finalName}-javadoc.ja
r"

basedir="${project.build.directory}/apidocs/"
                                     includes="**/**" />
                            </tasks>
                        </configuration>
                        <goals>
                            <goal>run</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>build-helper-maven-plugin</artifactId>
                <executions>
                    <execution>
                        <id>attach-artifacts</id>
                        <phase>package</phase>
```

```
                        <goals>
                            <goal>attach-artifact</goal>
                        </goals>
                        <configuration>
                            <artifacts>
                                <artifact>

<file>${project.build.directory}/${project.build.finalName}-javadoc.jar</f
ile>
                                    <type>jar</type>
                                    <classifier>javadoc</classifier>
                                </artifact>
                            </artifacts>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
```

```
</project>
```