Commentary on MaterialX implementation of OpenPBR

For discussion purposes, this document unpacks the MaterialX XML implementation of the existing Autodesk standard surface model, and the proposed OpenPBR model, into formulas which are bit easier to parse than the XML.

Then I give some initial commentary on the MaterialX implementations.

- Standard Surface repo
- OpenPBR repo
- MaterialX standard surface implementation

Contents

1 Autodesk standard surface implementation in MaterialX

1.1 Structure

The XML node structure can be reduced to the following formulas (where for simplicity, nodes which just reduce to algebraic operations are represented by formulas):

$$
\begin{aligned}
\text{shader constructor} = \textbf{surface}\Big(& \text{bsdf} &&= \text{coat\_layer},\\
& \text{edf} &&= \text{blended\_coat\_emission\_edf}\\
& \text{opacity} &&= \textbf{luminance}(\texttt{opacity})\Big)\\
\text{coat\_layer} = \textbf{layer}\Big(& \text{top} &&= \text{coat\_bsdf},\\
& \text{base} &&= \text{thin\_film\_layer} * (\texttt{coat\_color*coat} + (1 - \texttt{coat}))\Big)\,,\\
\text{thin\_film\_layer} = \textbf{layer}\Big(& \text{top} &&= \text{thin\_film\_bsdf},\\
& \text{base} &&= \text{metalness\_mix}\Big)\,,\\
\text{metalness\_mix} = \textbf{mix}\Big(& \text{fg} &&= \text{metal\_bsdf},\\
& \text{bg} &&= \text{specular\_layer},\\
& \text{mix} &&= \texttt{metalness}\Big)\,,\\
\text{specular\_layer} = \textbf{layer}\Big(& \text{top} &&= \text{specular\_bsdf},\\
& \text{base} &&= \text{transmission\_mix}\Big)\,,\\
\text{transmission\_mix} = \textbf{mix}\Big(& \text{fg} &&= \text{transmission\_bsdf},\\
& \text{bg} &&= \text{sheen\_layer},\\
& \text{mix} &&= \texttt{transmission}\Big)\,,\\
\text{sheen\_layer} = \textbf{layer}\Big(& \text{top} &&= \text{sheen\_bsdf},\\
& \text{base} &&= \text{subsurface\_mix})\\
\text{subsurface\_mix} = \textbf{mix}\Big(& \text{fg} &&= \text{selected\_subsurface\_bsdf},\\
& \text{bg} &&= \text{diffuse\_bsdf},\\
& \text{mix} &&= \texttt{subsurface}\Big)\,,\\
\text{selected\_subsurface\_bsdf} = \textbf{mix}\Big(& \text{fg} &&= \text{translucent\_bsdf},\\
& \text{bg} &&= \text{subsurface\_bsdf},\\
& \text{mix} &&= \texttt{thin\_walled}\Big)\,.
\end{aligned}
\tag{1}
$$

1.2 BSDFs

$$
\begin{aligned}
\text{coat\_bsdf} = \textbf{dielectric\_bsdf}(& \text{weight} &&= \texttt{coat},\\
& \text{tint} &&= (1, 1, 1),\\
& \text{ior} &&= \texttt{coat\_IOR},\\
& \text{roughness} &&= \text{coat\_roughness\_vector},\\
& \text{normal} &&= \texttt{coat\_normal},\\
& \text{tangent} &&= \text{coat\_tangent},\\
& \text{distribution} &&= \text{ggx},\\
& \text{scatter\_mode} &&= \text{R})
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
\text{translucent\_bsdf} = \textbf{translucent\_bsdf}(& \text{weight} &&= 1.0,\\
& \text{color} &&= \text{coat\_affected\_subsurface\_color},\\
& \text{normal} &&= \texttt{normal})
\end{aligned}
\tag{3}
$$

$$\begin{aligned}\text{diffuse\_bsdf} = \textbf{oren\_nayar\_diffuse\_bsdf}(&\ \text{weight} &&= \texttt{base},\\ &\ \text{color} &&= \text{coat\_affected\_diffuse\_color},\\ &\ \text{roughness} &&= \texttt{diffuse\_roughness},\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{4}$$

$$\begin{aligned}\text{subsurface\_bsdf} = \textbf{subsurface\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{color} &&= \text{coat\_affected\_subsurface\_color},\\ &\ \text{radius} &&= \texttt{subsurface\_radius * subsurface\_scale},\\ &\ \text{anisotropy} &&= \texttt{subsurface\_anisotropy})\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{5}$$

$$\begin{aligned}\text{sheen\_bsdf} = \textbf{sheen\_bsdf}(&\ \text{weight} &&= \texttt{sheen},\\ &\ \text{color} &&= \texttt{sheen\_color},\\ &\ \text{roughness} &&= \texttt{sheen\_roughness},\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{6}$$

$$\begin{aligned}\text{transmission\_bsdf} = \textbf{dielectric\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{tint} &&= \texttt{transmission\_color},\\ &\ \text{ior} &&= \texttt{specular\_IOR},\\ &\ \text{roughness} &&= \text{transmission\_roughness},\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\\ &\ \text{distribution} &&= \text{ggx},\\ &\ \text{scatter\_mode} &&= \text{T})\end{aligned} \tag{7}$$

$$\begin{aligned}\text{specular\_bsdf} = \textbf{dielectric\_bsdf}(&\ \text{weight} &&= \texttt{specular},\\ &\ \text{tint} &&= \texttt{specular\_color},\\ &\ \text{ior} &&= \texttt{specular\_IOR},\\ &\ \text{roughness} &&= \text{main\_roughness},\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\\ &\ \text{distribution} &&= \text{ggx},\\ &\ \text{scatter\_mode} &&= \text{R})\end{aligned} \tag{8}$$

$$\begin{aligned}\text{metal\_bsdf} = \textbf{conductor\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{ior} &&= \text{artistic\_ior. ior},\\ &\ \text{extinction} &&= \text{artistic\_ior. extinction},\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\\ &\ \text{distribution} &&= \text{ggx})\end{aligned}$$

$$\begin{aligned}\text{thin\_film\_bsdf} = \textbf{thin\_film\_bsdf}(&\ \text{thickness} &&= \texttt{thin\_film\_thickness},\\ &\ \text{ior} &&= \texttt{thin\_film\_IOR})\end{aligned} \tag{9}$$

2 OpenPBR (proposed) implementation in MaterialX

2.1 Structure

$$
\begin{aligned}
\text{shader constructor} = \mathbf{surface}\Big(\ & \text{bsdf} &&= \text{coat\_layer},\\
& \text{edf} &&= \text{blended\_coat\_emission\_edf}\\
& \text{opacity} &&= \mathbf{luminance}(\texttt{opacity})\Big)\\
\text{coat\_layer} = \mathbf{layer}\Big(\ & \text{top} &&= \text{coat\_bsdf},\\
& \text{base} &&= \text{thin\_film\_layer} * (\texttt{coat\_color*coat} + (1 - \texttt{coat}))\Big)\ ,\\
\text{thin\_film\_layer} = \mathbf{layer}\Big(\ & \text{top} &&= \text{thin\_film\_bsdf},\\
& \text{base} &&= \text{metalness\_mix}\Big)\ ,\\
\text{metalness\_mix} = \mathbf{mix}\Big(\ & \text{fg} &&= \text{metal\_bsdf},\\
& \text{bg} &&= \text{specular\_layer},\\
& \text{mix} &&= \texttt{metalness}\Big)\ ,\\
\text{specular\_layer} = \mathbf{layer}\Big(\ & \text{top} &&= \text{specular\_bsdf},\\
& \text{base} &&= \text{transmission\_mix}\Big)\ ,\\
\text{transmission\_mix} = \mathbf{mix}\Big(\ & \text{fg} &&= \text{transmission\_bsdf},\\
& \text{bg} &&= \text{fuzz\_layer},\\
& \text{mix} &&= \texttt{transmission}\Big)\ ,\\
\text{fuzz\_layer} = \mathbf{layer}\Big(\ & \text{top} &&= \text{sheen\_bsdf},\\
& \text{base} &&= \text{subsurface\_mix})\\
\text{subsurface\_mix} = \mathbf{mix}\Big(\ & \text{fg} &&= \text{selected\_subsurface\_bsdf},\\
& \text{bg} &&= \text{diffuse\_bsdf},\\
& \text{mix} &&= \texttt{subsurface}\Big)\ ,\\
\text{selected\_subsurface\_bsdf} = \mathbf{mix}\Big(\ & \text{fg} &&= \text{translucent\_bsdf},\\
& \text{bg} &&= \text{subsurface\_bsdf},\\
& \text{mix} &&= \texttt{thin\_walled}\Big)\ .
\end{aligned}
\tag{10}
$$

2.2 BSDFs

$$
\begin{aligned}
\text{coat\_bsdf} = \mathbf{dielectric\_bsdf}(\ & \text{weight} &&= \texttt{coat},\\
& \text{tint} &&= (1, 1, 1),\\
& \text{ior} &&= \texttt{coat\_IOR},\\
& \text{roughness} &&= \text{coat\_roughness\_vector},\\
& \text{normal} &&= \texttt{coat\_normal},\\
& \text{tangent} &&= \text{coat\_tangent},\\
& \text{distribution} &&= \text{ggx},\\
& \text{scatter\_mode} &&= \text{R})
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
\text{translucent\_bsdf} = \mathbf{translucent\_bsdf}(\ & \text{weight} &&= 1.0,\\
& \text{color} &&= \text{coat\_affected\_subsurface\_color},\\
& \text{normal} &&= \texttt{normal})
\end{aligned}
\tag{12}
$$

$$\begin{aligned}\text{diffuse\_bsdf} = \textbf{oren\_nayar\_diffuse\_bsdf}(&\ \text{weight} &&= \texttt{base},\\ &\ \text{color} &&= \text{coat\_affected\_diffuse\_color},\\ &\ \text{roughness} &&= \texttt{diffuse\_roughness},\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{13}$$

$$\begin{aligned}\text{subsurface\_bsdf} = \textbf{subsurface\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{color} &&= \text{coat\_affected\_subsurface\_color},\\ &\ \text{radius} &&= \texttt{subsurface\_radius * subsurface\_scale},\\ &\ \text{anisotropy} &&= \texttt{subsurface\_anisotropy})\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{14}$$

$$\begin{aligned}\text{sheen\_bsdf} = \textbf{sheen\_bsdf}(&\ \text{weight} &&= \texttt{fuzz},\\ &\ \text{color} &&= \texttt{fuzz\_color},\\ &\ \text{roughness} &&= \texttt{fuzz\_roughness},\\ &\ \text{normal} &&= \texttt{normal})\end{aligned} \tag{15}$$

$$\begin{aligned}\text{transmission\_bsdf} = \textbf{dielectric\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{tint} &&= \texttt{transmission\_color},\\ &\ \text{ior} &&= \texttt{specular\_IOR},\\ &\ \text{roughness} &&= \text{transmission\_roughness},\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\\ &\ \text{distribution} &&= \text{ggx},\\ &\ \text{scatter\_mode} &&= \text{T})\end{aligned} \tag{16}$$

$$\begin{aligned}\text{specular\_bsdf} = \textbf{dielectric\_bsdf}(&\ \text{weight} &&= \texttt{specular},\\ &\ \text{tint} &&= \texttt{specular\_color},\\ &\ \text{ior} &&= \texttt{specular\_IOR},\\ &\ \text{roughness} &&= \text{main\_roughness},\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\\ &\ \text{distribution} &&= \text{ggx},\\ &\ \text{scatter\_mode} &&= \text{R})\end{aligned} \tag{17}$$

$$\begin{aligned}\text{metal\_bsdf} = \textbf{generalized\_schlick\_bsdf}(&\ \text{weight} &&= 1.0,\\ &\ \text{color0} &&= \texttt{base * base\_color},\\ &\ \text{color90} &&= \texttt{specular * specular\_color},\\ &\ \text{roughness} &&= \text{main\_roughness}\\ &\ \text{normal} &&= \texttt{normal},\\ &\ \text{tangent} &&= \text{main\_tangent},\end{aligned}$$

$$\begin{aligned}\text{thin\_film\_bsdf} = \textbf{thin\_film\_bsdf}(&\ \text{thickness} &&= \texttt{thin\_film\_thickness},\\ &\ \text{ior} &&= \texttt{thin\_film\_IOR})\end{aligned} \tag{18}$$

3 Adjunct calculations in MaterialX implementations

3.1 EDF

$$\begin{aligned}\text{blended\_coat\_emission\_edf} = \textbf{mix}(&\ \text{fg} &&= \text{coat\_emission\_edf},\\ &\ \text{bg} &&= \textbf{uniform\_edf}(\text{color} = \texttt{emission * emission\_color}),\\ &\ \text{mix} &&= \texttt{coat})\end{aligned} \tag{19}$$

$$\begin{aligned}\text{coat\_emission\_edf} = \textbf{generalized\_schlick\_edf}(\ &\text{color0} &&= (1,1,1),\\ &\text{color90} &&= \text{coat\_ior\_to\_F0},\\ &\text{exponent} &&= 5.0,\\ &\text{base} &&= \text{coat\_tinted\_emission\_edf})\end{aligned} \tag{20}$$

$$\text{coat\_tinted\_emission\_edf} = \texttt{coat\_color} * \textbf{uniform\_edf}(\text{color} = \texttt{emission * emission\_color}) \tag{21}$$

3.2 Coat influence

$$\text{coat\_gamma} = 1.0 + \texttt{coat\_affect\_color*coat} \tag{22}$$

$$\text{coat\_affected\_subsurface\_color} = \text{pow}(\max(\texttt{subsurface\_color}, 0), \text{coat\_gamma})$$

$$\text{coat\_affected\_diffuse\_color} = \text{pow}(\max(\texttt{base\_color}, 0), \text{coat\_gamma})$$

3.3 Tangent rotation

$$\text{coat\_tangent} = \begin{cases}\textbf{normalize}(\text{coat\_tangent\_rotate}), & \text{if } \texttt{coat\_anisotropy} > 0\\ \texttt{tangent}, & \text{otherwise.}\end{cases} \tag{23}$$

$$\begin{aligned}\text{coat\_tangent\_rotate} = \textbf{rotate3d}(\ &\texttt{tangent},\\ &\text{amount} &&= 360 * \texttt{coat\_rotation},\\ &\text{axis} &&= \texttt{coat\_normal})\end{aligned} \tag{24}$$

$$\text{main\_tangent} = \begin{cases}\textbf{normalize}(\text{tangent\_rotate}), & \text{if } \texttt{specular\_anisotropy} > 0\\ \texttt{tangent}, & \text{otherwise.}\end{cases} \tag{25}$$

$$\begin{aligned}\text{tangent\_rotate} = \textbf{rotate3d}(\ &\texttt{tangent},\\ &\text{amount} &&= 360 * \texttt{specular\_rotation},\\ &\text{axis} &&= \texttt{normal})\end{aligned} \tag{26}$$

3.4 IOR remapping

$$\begin{aligned}\text{artistic\_ior} = \textbf{artistic\_ior}(\ &\text{reflectivity} &&= \texttt{base\_color * base},\\ &\text{edge\_color} &&= \texttt{specular\_color * specular})\end{aligned} \tag{27}$$

$$\text{coat\_ior\_to\_F0} = \left(\frac{1 - \texttt{coat\_IOR}}{1 + \texttt{coat\_IOR}}\right)^2 \tag{28}$$

3.5 Roughnesses

$$\begin{aligned}\text{main\_roughness} = \textbf{roughness\_anisotropy}(\ &\text{roughness} &&= \text{coat\_affected\_roughness},\\ &\text{anisotropy} &&= \texttt{specular\_anisotropy})\end{aligned} \tag{29}$$

$$\begin{aligned}\text{coat\_affected\_roughness} = \textbf{mix}(\ &\text{fg} &&= 1.0,\\ &\text{bg} &&= \texttt{specular\_roughness},\\ &\text{mix} &&= \texttt{coat\_roughness * coat\_affect\_roughness * coat})\end{aligned} \tag{30}$$

$$\begin{aligned}\text{transmission\_roughness} = \textbf{roughness\_anisotropy}(\ &\text{roughness} &&= \text{coat\_affected\_transmission\_roughness},\\ &\text{anisotropy} &&= \texttt{specular\_anisotropy})\end{aligned} \tag{31}$$

$$\text{coat\_affected\_transmission\_roughness} = \mathbf{mix}(\ \text{fg} = 1.0,\ \text{bg} = \text{transmission\_roughness\_clamped},\ \text{mix} = \texttt{coat\_roughness * coat\_affect\_roughness * coat}) \quad (32$$

$$\text{transmission\_roughness\_clamped} = \mathbf{clamp}(\texttt{specular\_roughness + transmission\_extra\_roughness}) \quad (33)$$

4 MaterialX commentary in context of OpenPBR

4.1 Commentary on overall differences in formalism

- Generally, the existing MaterialX implementation seems to be a fairly faithful transcription of the formulas in the Autodesk Standard Surface spec into an XML node form. But this then inherits the assumption of that spec that the model is represented by a linear combination of weighted BSDFs, which is not assumed in OpenPBR where the spec attempts to describe an unambiguous physical structure, remaining agnostic about how this is implemented/approximated. This may be fine as MaterialX will amount to one particular approximate implementation (possibly the canonical/standard one that closes the gap between the physical description and a practical implementation), but we need to be careful about how this approximation is arrived at since it is no longer explicit in the spec, and make sure that is is reasonably faithful to the form of the underlying physical model.

- The MaterialX model consists of BSDFs composed by **layer** and **mix** operations. There is no explicit mention of media/"VDF". For example subsurface_bsdf in MaterialX contains the parameters of the subsurface medium, which is not a BSDF. In OpenPBR, there is an explicit distinction between the interface of the medium (a dielectric BSDF), and the medium itself (a "VDF", borrowing the MaterialX terminology).

- The concept of BSDFs/closures having a "weight" is not generally present in OpenPBR, as it doesn't necessarily make sense since not all BSDF models have a simple albedo scale factor as one of their parameters. If a weight parameter is present which is supposed to scale the albedo of the BSDF this is said explicitly (e.g. `base_color` parameters scales the diffuse BSDF albedo, which makes good sense since albedo is a parameter of diffuse BSDFs). The `specular` weight and color will be interpreted in OpenPBR as a (non-physical, but artistically useful/important) multiplier of the Fresnel factor, rather than a generic weight. Some weights presented in MaterialX as a BSDF weight operate in OpenPBR by controlling the coverage/presence of a layer (e.g. `sheen`).

- OpenPBR introduces the concept of a layer "coverage" or "presence" weight, which is the mix weight between the coated and uncoated substrate. This is essentially what the coat weight in Standard Surface is doing, via a lerp formula. In MaterialX, this coverage weight is not really explicit instead the base of the coat is "multiplied" by the lerp formula. This form of operation made sense in the context of a model which is blending BSDFs (such as standard surface), but doesn't make sense in OpenPBR where one is composing slabs of material not BSDF functions. It could be clearer to augment the MaterialX **layer** operator with a coverage weight (which is trivial to transform into the lerp formula form if working with BSDF linear combinations).

- The **layer** operation takes a top and a base in MaterialX. Presumably "top" means the BSDF of the top interface, and "base" means the (BSDF of the?) material substrate on which the layer is bonded. But in general there will be some media in the sandwich between the interfaces (or at the bottom of the layer stack), which is not explicit in the MaterialX representation. In OpenPBR we try to make this explicit by representing the material as consisting of physical slabs with defined interface BSDFs and internal medium, denoted $\text{Slab}(f^t, V, f^b)$. If there is a "layer" operator in the formalism, it probably should allow explicitly for the presence of the media, otherwise accounting for media requires "hacks" such as the manual scaling factor in the base of the MaterialX coat layer.

- In OpenPBR there is no distinction between transmission_bsdf and specular_bsdf (i.e. two separate BSDFs with one half-space artificially ignored via a "scatter mode"), these are just the reflected/transmitted lobes of the single dielectric BSDF (bounding the underlying medium). Any non-physical/artistic multipliers of the dielectric's reflected/transmitted lobes (and how the energy balance is affected) will be made explicit in the description of this single BSDF. Again, this is

not properly described yet in the OpenPBR spec. It could be reasonable for MaterialX to use this scatter mode notion, but arguably it is more likely that renderers will implement a single dielectric BSDF with logic to handle the balance between reflection and transmission.

- As noted, in OpenPBR there will be a single dielectric BSDF representing the interface above the medium described either by subsurface, transmission, or diffuse parametrizations. The translucent_bsdf in MaterialX seems to be representing the base medium corresponding to the transmission properties, but:
  - It's not explicit that this is a dielectric BSDF
  - The volumetric parameters are missing (are those outside the scope of the existing MaterialX formalism?)
  - This is presumably supposed to be a model of a medium (VDF), not a BSDF interface, so the name translucent_bsdf is misleading (as is subsurface_bsdf).

4.2 Commentary on some more technical discrepancies in model

- Note that sheen is being moved to the top of the material in OpenPBR, not just renamed to fuzz. Also the sheen layering operation is not properly transcribed from the Standard Surface spec, as the sheen color should not be included in the reflectance for albedo scaling (otherwise the complementary color would tint the underlying base). We will need to rationalize this in the OpenPBR spec as corresponding to the sheen medium having a non-colored/grey extinction.

- In OpenPBR, the primary specular lobe will no longer have the interpretation as an explicit “specular layer”, instead it is just due to the dielectric BSDF of the boundary of the base dielectric. To be fair, this is not properly fleshed out yet in the OpenPBR draft spec.

- In MaterialX, thin film is represented as an explicit layer (of dielectric?) under the coat. In practice, in Arnold this is handled by modifying the Fresnel factor of the underlying dielectric and metal BSDFs. It would not make sense for example to treat this as a regular layer composed by e.g. albedo scaling, as the relevant physical effect is a special one requiring wave optics to solve, so seems more appropriate to handle these details within the black-box BSDF implementations than as part of the material layering formalism. This is still an open question in OpenPBR though.

- In MaterialX, the **mix** operation is overloaded to mean both statistical mix of materials, as well as a blend of EDFs, and a regular lerp of numeric quantities. This seems reasonable as a convenience, though potentially slightly confusing.

- Minor point, but having an explicit “foreground” and “background” in the **mix** operation doesn't quite seem an appropriate analogy, as the two components are on an equal footing, the only question is which component the weight controls (so the form $\mathbf{mix}(S_0, S_1, w_1)$ seems clearer).

- Just to note, the formulas (in Standard Surface, transcribed into MaterialX) that implement the effect of the coat on the diffuse/subsurface color and roughnesses are one particular attempt at a rough approximation of effects that would happen automatically in a physically accurate simulation of the light transport. Though it may be fine for MaterialX to use these particular rough approximations explicitly, we're unclear whether to include such specific approximations in the OpenPBR spec.

- In MaterialX, there is a single EDF associated with the entire surface, so the effects supposed to be due to the emission occurring under the coat have to be inserted by hand. It would be more explicit to associate the EDF with a particular “slab” in the OpenPBR formalism (i.e. the entire material below the coat).

- The explicit formulas defined for the tangent rotation seem perhaps unnecessary, as any implementation will probably have their own routines to do this (given some unambiguous definition of the meaning of the parameters), rather than executing it via the node graph.

- The use of generalized_schlick_edf for the emission seems quite specific, and unclear what setting $\mathrm{color90} = \mathrm{coat\_ior\_to\_F0}$ is trying to approximate.