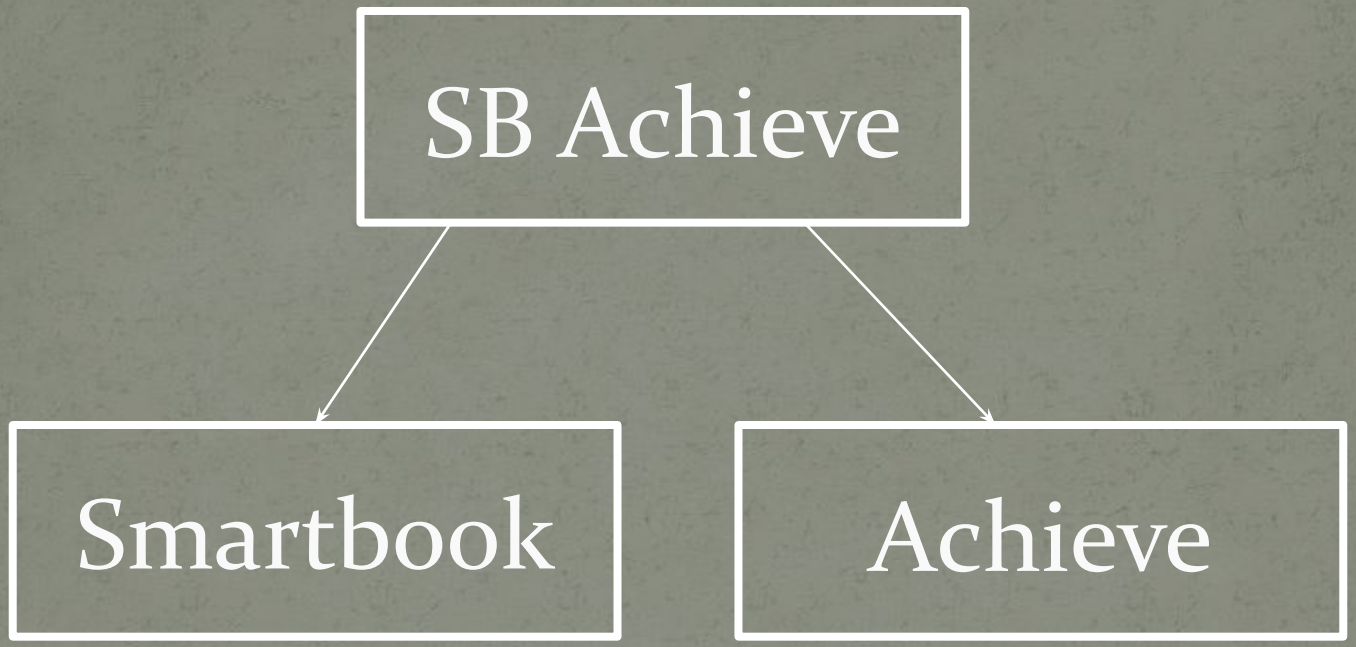
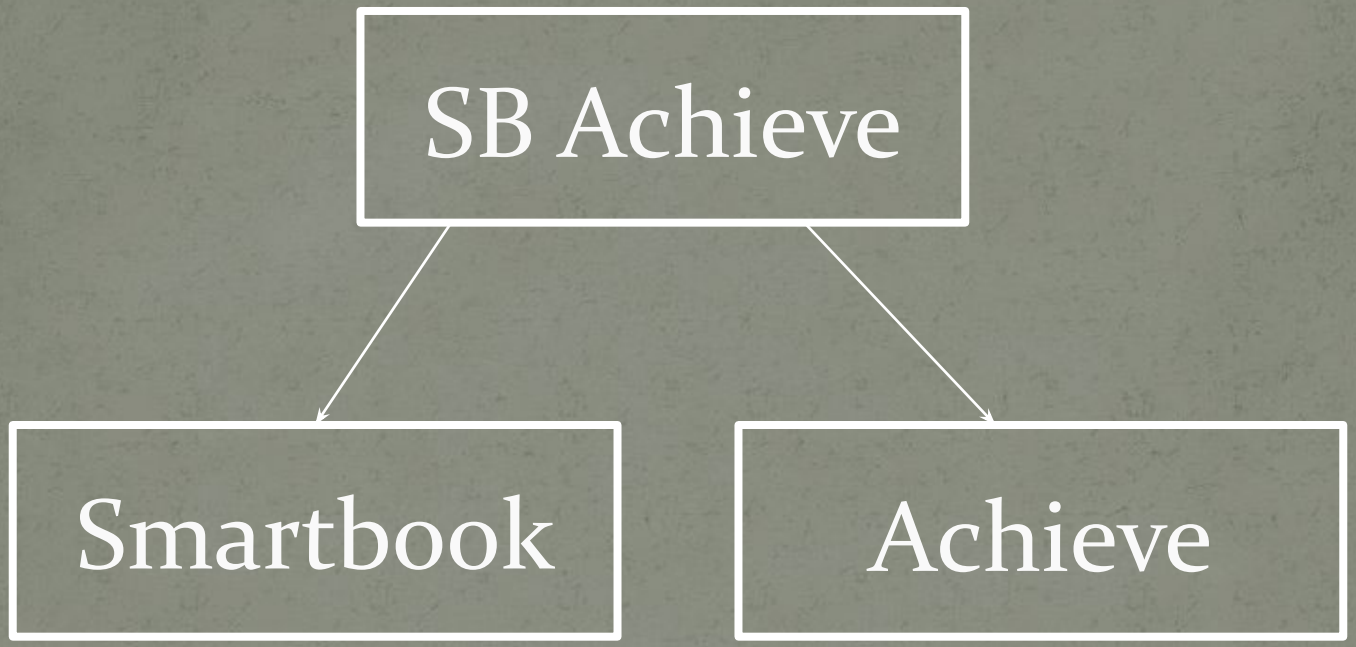
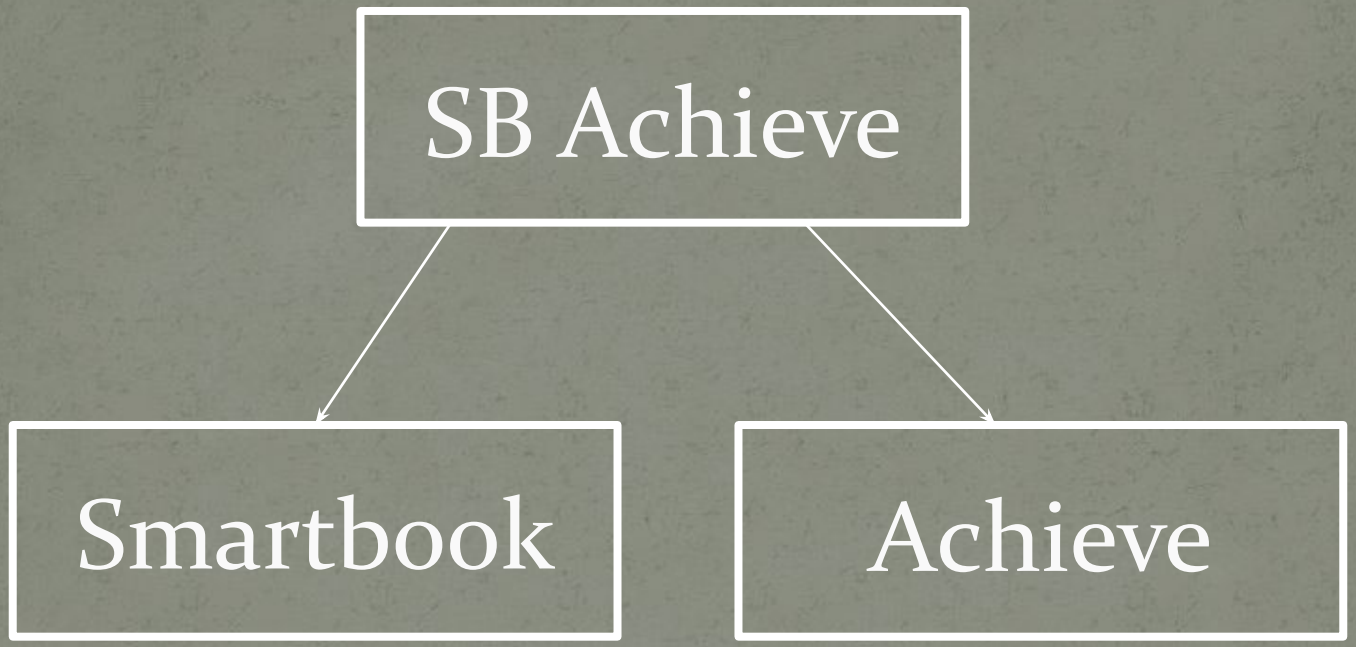
SB/LS Achieve (Master) Introduction


SB Achieve
Smartbook
Achieve



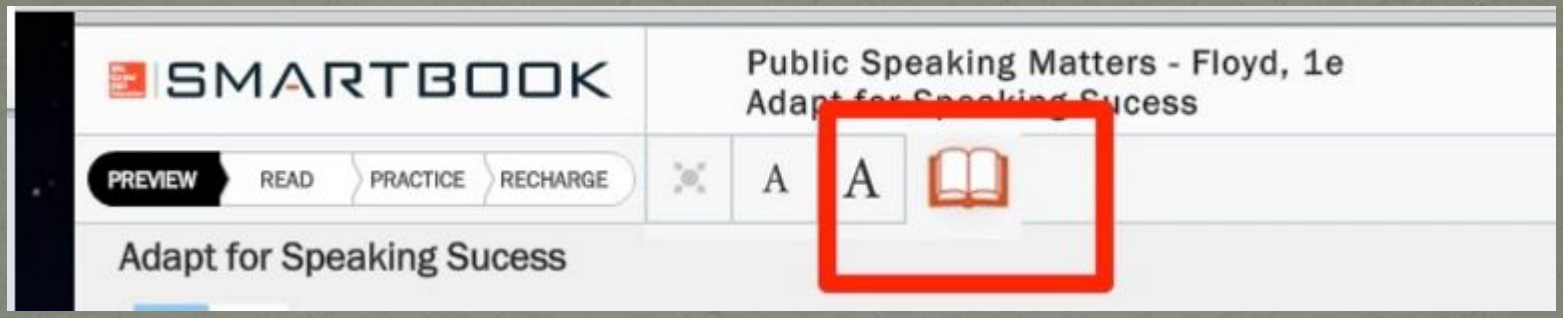
Good known!



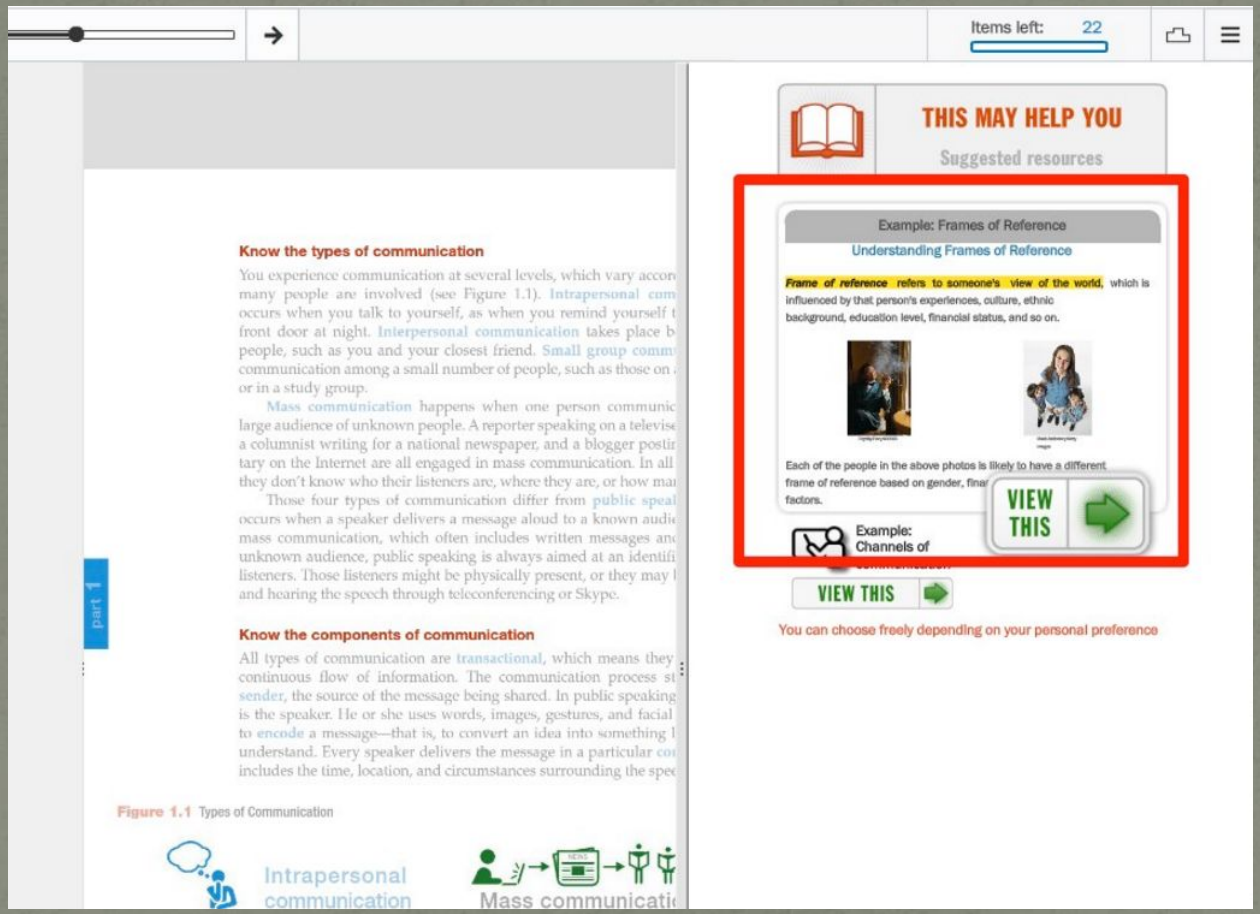
Good known!

???

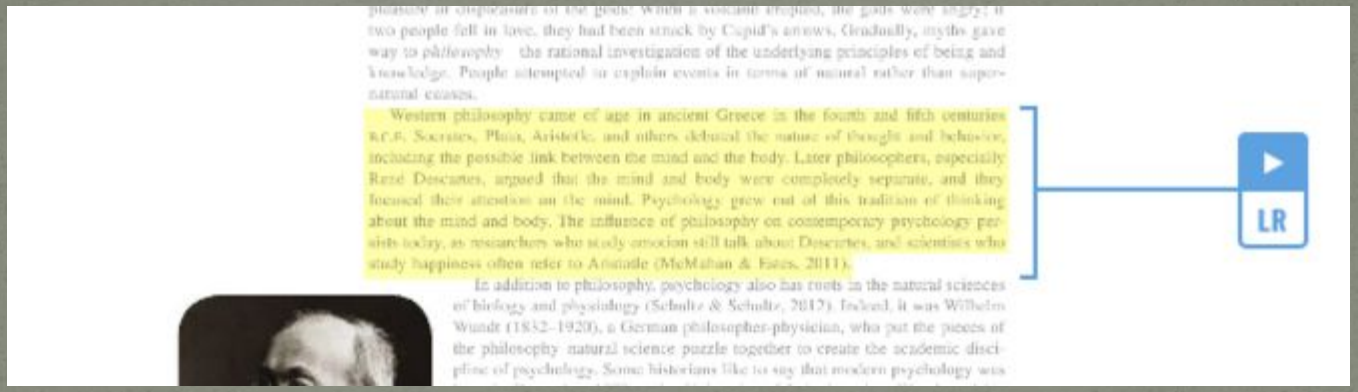
Old Panel for SB LRs



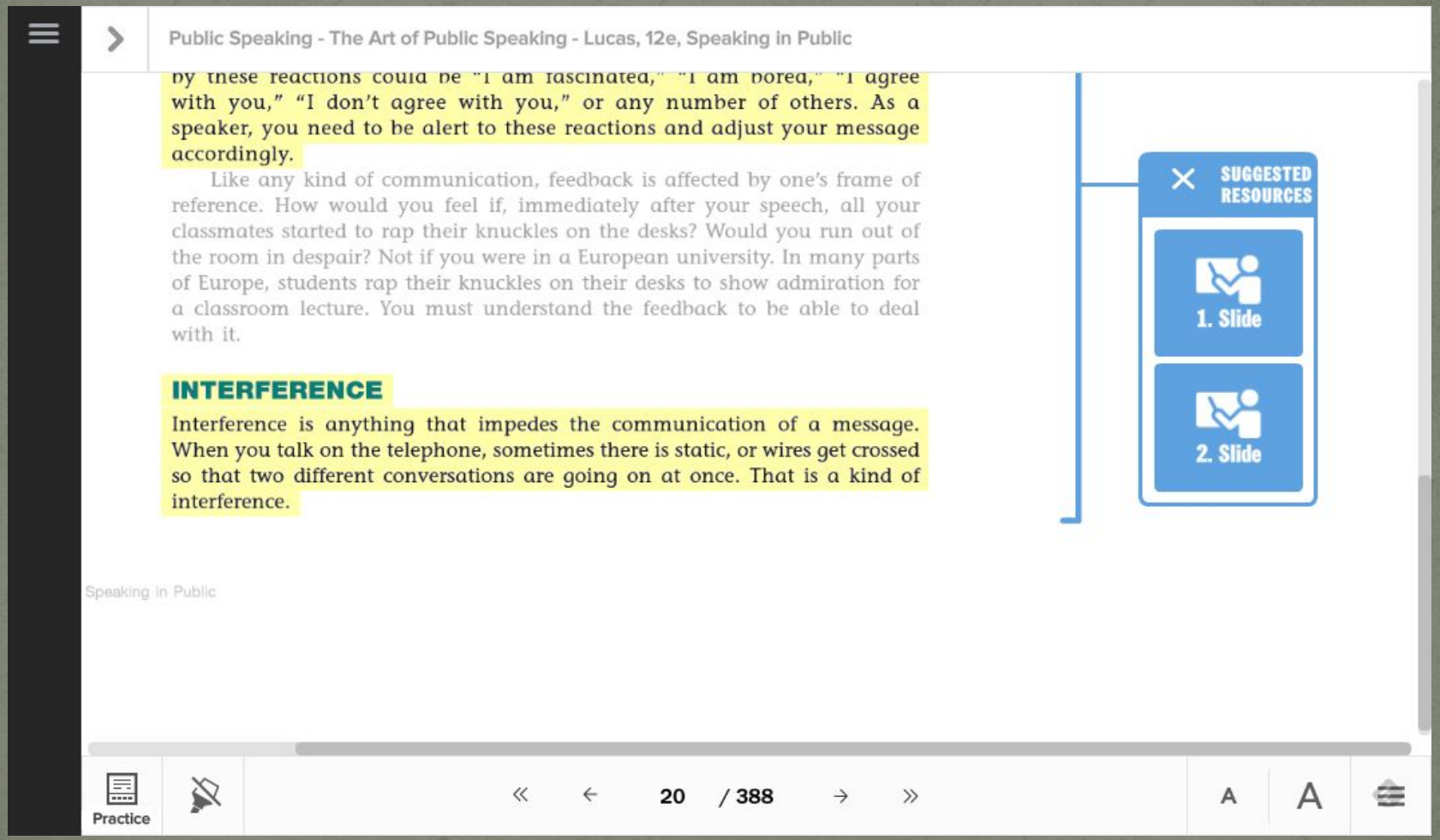
Old Panel for SB LRs

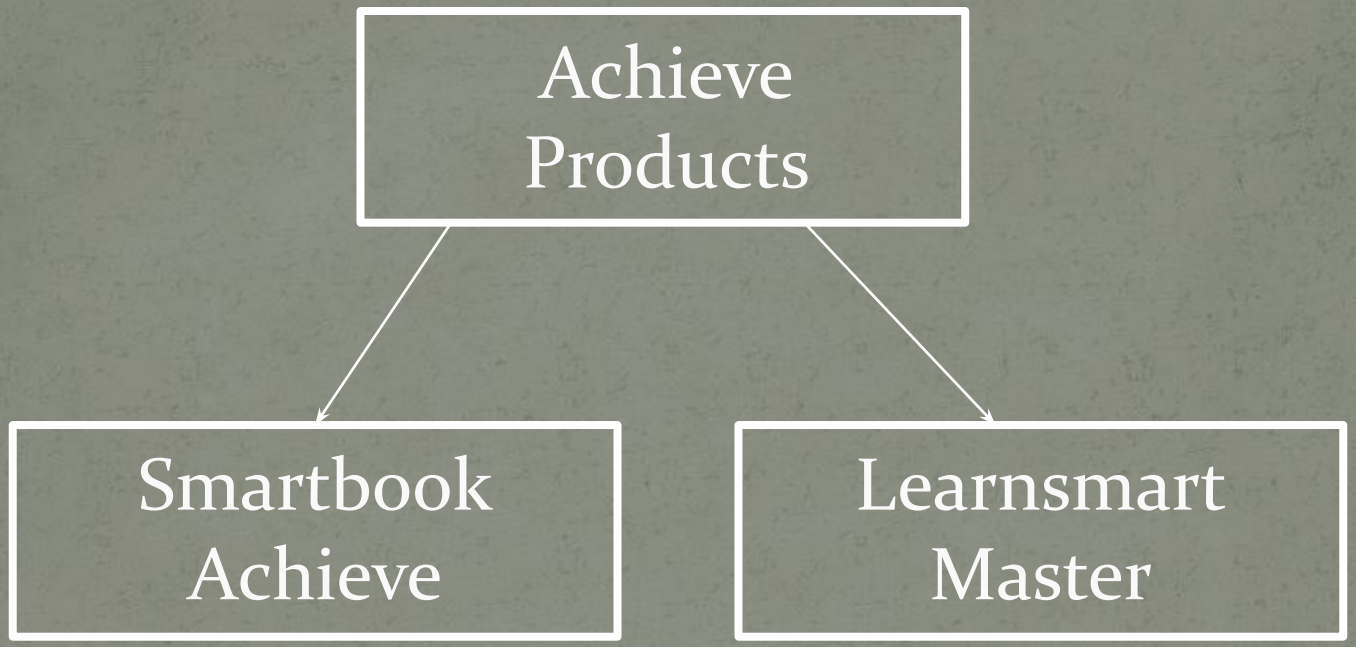


New Panel for SB LRs

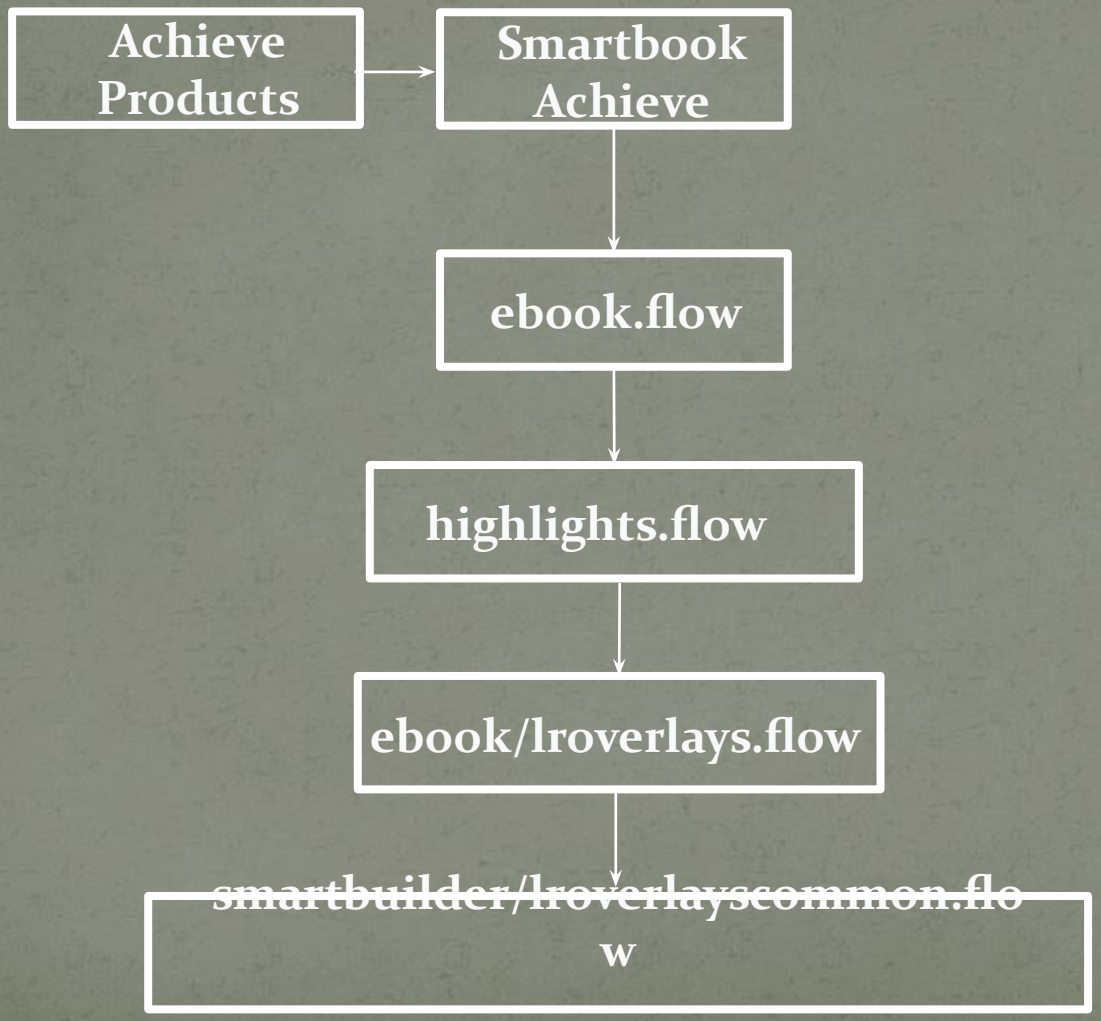


New Panel for SB LRs

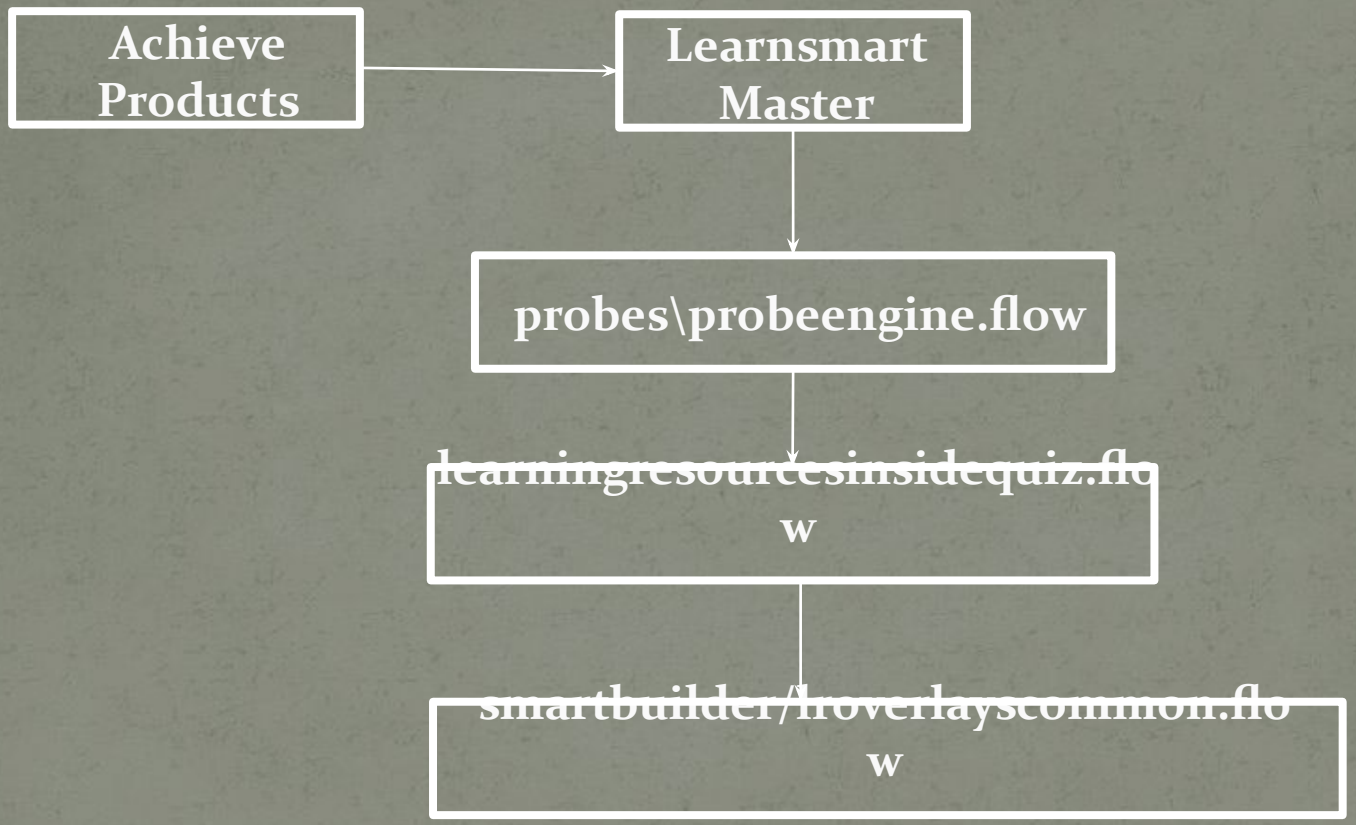



Achieve
Products
Smartbook
Achieve
Learnsmart
Master

Smartbook Achieve file dependencies



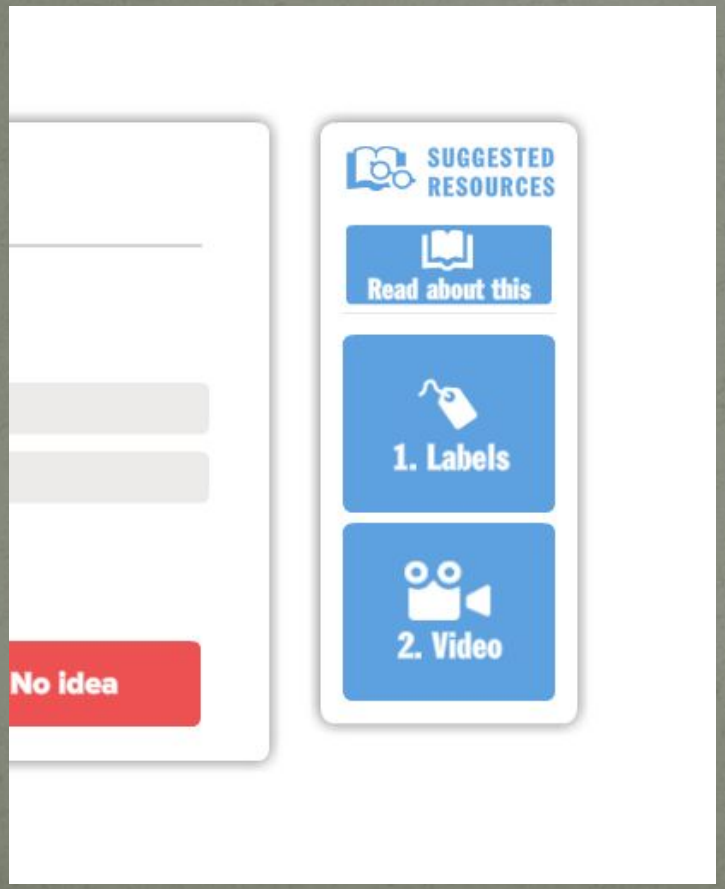
Learnsmart Master file dependencies



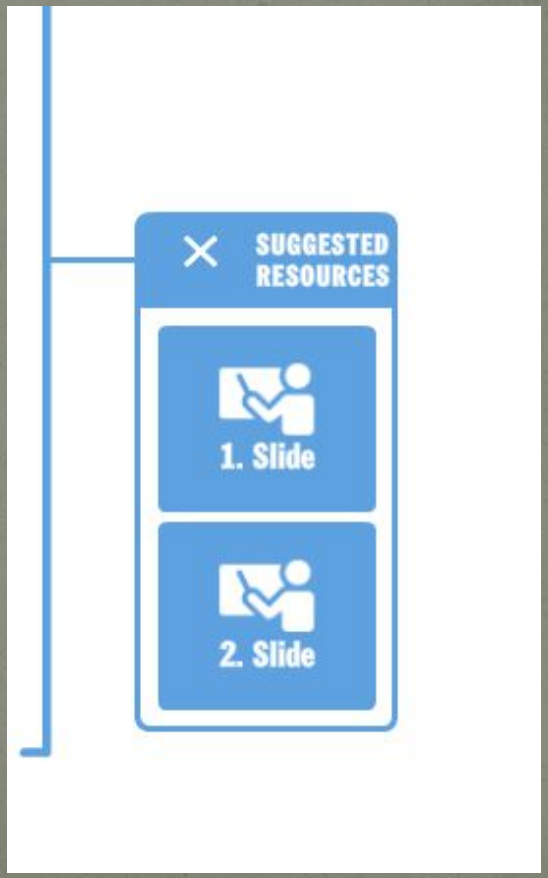
smartbuilder/lroverlayscommon.flow

Learnsmart Master vs Smartbook LRs Panel

Master



SB Read mode



Cases where LRs list is used:

- Smartbook Achieve: reading mode;
- Smartbook Achieve: practice mode;
- Learnsmart Master

lroverlayscommon.flow

Overview

```
getLrListForm(
    lrsB : Behaviour<[LearningResource]>,
    mouseOnArrayB : Maybe<[DynamicBehaviour<bool>]>,
    checkmarksArrayB : ref [DynamicBehaviour<bool>],
    makeOpenLrFn : (int, LearningResource) -> ((Form) -> (()->void)),
    continueFn : () -> void,
    ovSizes : lrOverlayButtonSizes,
    strokeOn : bool,
    forSmartBook : bool,
    getCustomScores : Maybe<ref () -> Maybe<[Pair<string, double>]> >,
    footerEnabled : bool) -> Form;
```

lroverlays.flow

Overview

```
export{

    lrEbookStyled(x0 : double, y0 : double, h : double, lrs : [lrWithMouseB]);
    overlaysIntersect : (lrEbookStyled, lrEbookStyled) -> bool;
    mergeOverlays : (lrEbookStyled, lrEbookStyled) -> lrEbookStyled;

    overlaysLayers : ref [DynamicBehaviour<Form>] = ref [];
    overlaysClosedB : ref [ref [DynamicBehaviour<bool>]] = ref [ref []];

    lrPreviewPanel : DynamicBehaviour<Form> = make(Empty());
    lrPanelVisible : DynamicBehaviour<int> = make(0); //[-Explain-] What values could be and what are they mean

    setClosedBehaviourByPage : (int, bool) -> void;

    makeOverlayForm(index : int, page : int, overlay : lrEbookStyled) -> Form;
    makeOverlayFormForProbe(LRs, probeCalloutFn_, addProbeShadow_) -> Form;
    overlaysProbePanel = makeEmpty();
    overlaysSetWindowFn : ref (Form) -> void = ref \form -> {};
```

highlights.flow

Overview

```
buildHighlightLayer : (highlightBlocks : [HighlightedRectBlock],
                       pictures : [HighlightedRectBlock],
                       isMouseDownOnHighlight : ref bool,
                       showSideOverlay : DynamicBehaviour<int>,
                       page : int)->Form;

buildHighlightLayer(highlightBlocks, pictures, isMouseDownOnHighlight, showSideOverlay, page){

    opacity = make(1.0);

    if (isContainerSkin())
        bidirectionalLink(curHlOpacity, opacity, idfn, idfn) |> ignore
```

highlights.flow

Overview

```
mouseOnLR = if(hasProductOption(SmartBookWithAchieve())){
    rightMaxX = fold(highlightBlocks, 0.0, \acc, block -> {...
    });

    loHighlightOn : ref [Behaviour<bool>] = ref [];

    overlays = fold(highlightBlocks, [], \acc : [ref lrEbookStyled], block -> {...
    });

    compFn = \x : ref lrEbookStyled, y : ref lrEbookStyled -> if (^x.y0 < ^y.y0) -1 else 1;
    overlaysSorted = mergeSort(overlays, compFn);

    len = length(overlaysSorted);
    overlaysFiltered = foldi(overlaysSorted, [], \i, acc : [ref lrEbookStyled], item -> {...
    });

    closed = map(overlaysFiltered, \ov -> make(true));
    (^overlaysClosedB)[page] := closed;

    overlaysForm = Group(
        mapi(overlaysFiltered, \i, ov -> makeOverlayForm(i, page, ^ov))
    );

    nextDistinct((^overlaysLayers)[page], Select(currentHighlightMode, \v -> If (v == ReadingHighlightMode, overlaysForm)));

    Some(merge(^loHighlightOn))
} else {
    None()
};
```

highlights.flow

Overview

```
buildHighlightsForPage(highightsBlocks, picturesBlocks, headingsBlocks, page, isMouseDownOnHighlight, showSideOverlay){
    hblocks : [HighlightedRectBlock] =
        getHighlightedRectBlocksForPage(highightsBlocks, page);
    allBlocks : [HighlightedRectBlock] = concat(
        filter(headingsBlocks, \block -> block.pageId == page),
        hblocks
    );

    // HighlightedRectBlock : (pageId : int, loid : string, rects : [CropRect]);
    // CropRect : (left : double, top : double, right : double, bottom : double);
    form = buildHighlightLayer(allBlocks, filter(picturesBlocks, \picture->picture.pageId == page), isMouseDownOnHighlight,

    allRects = concatA(map(allBlocks, \ b -> b.rects));

    crect = if (length(allRects) == 0) None() else {

        // We already know that array is not empty here
        e0mina = \ a -> either(minA(a), 0.0);
        e0maxa = \ a -> either(maxA(a), 0.0);
        minLeft   = map(allRects, \ r -> r.left  ) |> e0mina;
```

highlights.flow
Overview

If we will look furher up then we will go through the next chain of functions:

- buildHighlightsForPage
- calculateAndDisplayHighlightsForLos
- displayHighlights

The last one then is exported and used into the ebook.flow (initPdfFrame fn).

ebook.flow

Overview

```
buildPageBlock(
    size : WidthHeight, pdf : EPdf, page : int,
    hightlightLayer : Form,
    dimRects : Behaviour<[HighlightedRectBlock]>,
    undimRects : Behaviour<[HighlightedRectBlock]>,
    headsRects : Behaviour<[HighlightedRectBlock]>,
    onPicturesLoaded : ()->void,
    onPicturesLoadingError : () -> void)
{
    if (pdf != emptyPdf) {
```

ebook.flow
Overview

```
overlaysAddedWidth = if (hasProductOption(SmartBookWithAchieve()) && cropPresented(pdf)) 320.0 else 0.0;
sizeB = select(currentHighlightMode, \chm -> WidthHeight(size.width + if (chm == SpreadHighlightMode) 0.0 else o
tmpFrm = Size2(sizeB,
    if (cropPresented(pdf)) {
        _x = make(max(pdf.crop.left, 0.0)); _y = make(pdf.crop.top);

        mow = make(0.0);
        mutoverlays = Inspect(
            [Width(mow)],
            if (existsIndex(^overlaysLayers, page - 1)) Mutable(^(overlaysLayers)[page-1]) else Empty()
        );

        Group([
            Crop(_x, _y, _w, _h, mergedLayers) |> debugColorFrame(0xff00ff),
            Align(0.0, 0.0,
                Translate(
                    select2(_x, mow, \ x, w -> {
                        -x - 20.0 + pdf.pageWidth - w
                    }),
                    select(_y, \y -> -y),
                    mutoverlays |> debugColorFrame(0)
                ) |> debugFrame
            )
        ])
    } else Group([mergedLayers, Align(1.0, 0.0, if (existsIndex(^overlaysLayers, page - 1)) Mutable(^(overlaysLa
) |> debugColorFrame(0x008800);
```

ebook.flow
Overview

```
selectPDF(pdf : EPdf) -> void {
    setCurrentEPdf(pdf);
```

```
    highlightLayers := [];
    overlaysLayers := [];
    overlaysClosedB := [];
    highlightBounds := [];

    next(offlinePreloadingRun, false);
    fori(1, pdf.pagesCount, \_i -> {
        //blockTimes := arrayPush(^blockTimes, []);
        highlightLayers := arrayPush(^highlightLayers, make(Empty()
            //Rectangle(2000.0, 2000.0, [Stroke(0xC7F5CF), StrokeWidth(3.0), Fill(0xC7F5CF)])
        ));
        overlaysLayers := arrayPush(^overlaysLayers, make(Empty()));
        overlaysClosedB := arrayPush(^overlaysClosedB, ref []);
        highlightBounds := arrayPush(^highlightBounds, make(None()));

        v = buildPageBlock(getPageSize(), pdf, _i,
            Mutable((^highlightLayers)[_i-1]),
```

Learnsmart Master

Overview

1. In the **learningresourcesinsidequiz.flow** file the decorator for the LRs list is formed;

2. It is applied in the **probeengine.flow** using the lrsInsideQuiz functions.

SB Achieve in the Practice mode

Overview

1. Looks almost the same as in the Learnsmart Master; Decorator for LRs list is formed in the **lroverlays.flow** in makeOverlayFormForProbe fn;

2. It is applied in the **smartbook\smartbookassignment.flow**.

Thanks for attention!

Q & A