

Justin North (00:00):

Hello everyone. My name is Justin North. I'm the Vice President of Technology at IGI Global. I've brought with me today Clayton George. Clayton, your full title is director of, I do apologise...

Clayton George (00:12):

Director of the workflow.

Justin North (00:13):

The workflow, we'll call it that. I had it on the slide as a note, but it is different slide deck. I do apologise there. So our case study is really around automation and centralization, and just to give you a little background of where, I think Chand just mentioned that we've been a Typefi customer for a very long time. That's true. It was actually implemented at IGI Global prior to the IT department being created. The IT department actually was designed and brought individuals in in 2009. So we came in after the implementation. So what we'll have in this case study is a description of the initial implementation followed by where we're at today and where we're going to be in the future. So the way that we see it, we had the original implementation in 2008 and an upgrade that's so drastic that it's basically another implementation happening right now. So we'll go ahead and get started. So just a little information about IGI Global, who we are. It was founded in 1988. We're a mid-size publisher, probably a little bit less than 50 employees and we're located in Hershey, Pennsylvania.

(01:24):

The industry that we're in, we publish research-based academic books and journals with our core subjects being business and management, science, technology and medical, and then education. The products that we provide are reference books, reference journals. We also do case books and authored books and things of that nature. But our real bread and butter is reference books. Now, these reference books aren't books that you're probably used to. We're talking about maybe three or four editors with 15 chapters. Each chapter could have anywhere between one and 12 contributors. So as you can see, trying to keep up with all these individuals is very difficult just because of the volume of individuals that can be associated with one piece of material. And the formats that we publish in. We do hardcover, soft cover, which obviously involves Typefi, PDF eBooks, EPUB collections, and when I say collections, it's really a collaboration of books and journals that we sell, whether it be on our proprietary platform or through our partners. And then on demand, which are just chapters and articles that are sellable on our bookstore for researchers to purchase individually.

(02:42):

So what I wanted to do is I wanted to provide a description of what the environment at IGI Global looked like prior to bringing in any technology. So this would be prior to the Typefi implementation. The way that the workflow worked, everything was pretty much by hand, even from the project management, it was completely a manual process. Individuals were keeping track of the project in spreadsheets, in emails, on sheets of paper. It was very antiquated. The typesetting was completely 100% manual, copy from a Word document, put it in InDesign, typeset it and move on. The consistency was a bit of a problem because as you know, if you're manually doing something and you have five people doing it, all five people are going to do it differently. So there was zero consistency across the products. Obviously it's very time consuming to do it manually. Very tedious work and it was primarily print centric. It was almost exclusively print centric. So that was the environment prior to bringing in Typefi. And I do apologise, I can't scroll here.

(04:02):

We'll just move on there. So what I wanted to do then is just talk about what that implementation in 2008 looked like. We were bringing in a content management system, which was there to not only do the

content management, but also manage the workflow of the projects after acquisition. We brought in eXtypes, obviously we mentioned Inera earlier, to provide NLM XML, and Typefi, and bring in Typefi to do the typesetting in InDesign. Once again, this was an implementation back in 2008. So what was performed after the IT department was brought in. What we did is a post-implementation assessment, identifying the pros and the cons of this implementation. Obviously, the pros were obvious, the typesetting—no longer manual, increased output and consistency, not to mention the XML, which is beneficial for obvious reasons. The cons were the workflow management. The workflow management that we brought in just wasn't working for us.

(05:02):

We had all these benefits, the benefits of the XML, the benefits of the typesetting, the consistency, but we weren't really gaining output because of the workflow management, we were struggling with it. It was like a constant battle. And then also the fact that we were still very print centric. Even though we had automation, we had technology, the primary focus of the organisation was the print side. So basically what we needed to do as an IT department was identify ways to address these gaps, if you will, or these voids in our workflow or what was holding us back from really producing content and pushing it out at a high volume. So this once again is at the end of the original implementation. So I'd like to bring Clay in so that Clay can describe how we filled the gaps and how we got to where we were six months ago. And then we'll talk about the upcoming or existing upgrade that we're currently working on.

Clayton George (06:08):

Thank you. You'll have to forgive. The last time I was public speaking, I believe was high school about 16 years ago. So we'll just do our best and move forward. So I arrived at IGI Global and just got hired into the IT department and we had a problem where we were having Word documents and they were getting converted into this really nice XML and then we were taking that and putting it through a content management system. And the management system would give us errors that were hard to find, they were hard to fix, hard to understand. And then eventually we'd get it through there and it'd get the Typefi and we'd get this really great InDesign out of it. The problem was the CMS. So Justin tasked me as a new developer to try and eliminate that problem or replace it. And so what we ended up with was a proprietary application that internally we referenced as the PUB, stands for production utility box, because Batman had recently come out.

(07:23):

And so the big benefits of this is we were able to integrate the PUB with our shared backend database. The database runs our website, the database runs our extranet, the database runs our submission system, which is the E Editorial Discovery. And we were able to use our folder structure as a proprietary CMS through using the PUB app, allowing our end users to still use the standard Windows file explorer, but also manage the files through the PUB app, through some fun permissions and things like that. And then we had a problem existing still with, I call it the book level files here. It was like in the InDesign book. Back, we're talking back CS3, CS4 kind of era. The InDesign books were not working for us, in the same way, at the scale that we were doing. We're trying to push 10, 15 books a week with a staff of seven, eight people, and these books are anywhere between 10 to 15, 20 chapters. And then we're trying to do edits, re-pagination and it just wasn't working. So through the PUB app we developed something that I refer to as the spine file, the spine being the part of the book that holds the book together. So the spine file is the part of the file that holds all the documents together and we were able to eliminate the InDesign book file from our entire workflow.

(09:16):

Then using all the metadata from the spine file that I was able to create, we were able to generate our own TOCs, front and back material, et cetera. That's just a brief rundown of what the PUB app was able to do. There's other things, but that's just a good overview. And then we get into working with Typefi. So when

I was tasked with this, the real thing I needed to do is get from the NLM we were getting from eXtyle and get it into InDesign. And so how do we do that, we were able to integrate with the Typefi API with just reading the documentation. I believe Justin and I read over the documentation, consulted a little bit with Caleb, and we were able to get it up and running in I think a week to two weeks, we had it in production and they were producing InDesign files instead of using the CMS.

(10:20):

The job management was great. It was easy to send jobs off, let 'em go overnight, come back the next morning, download the InDesign. Everything was easy to track. And then the job options, we have eight and a half by 11 books. We have eight and a half by 11 encyclopaedias. We have authored books, we have seven by 10, we have seven by 10 authored. And all of this stuff is just easy to manage as separate projects in Typefi. You want this, you just put it through this workflow and you got it out the other end in the right size and the right styling. The template flexibility was what really made the PUB app's future automation in a possibility. I got tasked with generating the TOC because the old CS4 TOC creation was just not working for us. I got tasked with creating front and back material, the preface, book series page, title page, all these things that existed, the data existed in the database, but how do we get it through to be an InDesign document?

(11:31):

And I'm a very naturally curious person, as I feel most developers are or should be. And Justin was just like, hey, try and figure this out. So my first route was, how do I do a comm—I was already interacting with InDesign a little bit through a comm interop, automating InDesign through C# code, and I was like, can I just create an entire InDesign document through the interop and place all the elements myself and just figure it out? And I spent maybe a week trying to do that and I was like, this is a nightmare. And so I looked at Typefi, I'm like, well, this is dumb. We already have a solution that's doing this for us. Sometimes you got to take a weekend, look at your current time and figure it out. So I looked at Typefi and I was like, how is Typefi doing this?

(12:28):

And so I saw, well, they're taking the NLM and converting it to CXML. Okay, what's CXML? Content XML. So I read the documentation on Content XML. I was like, well, this is great. And then I saw the relationships between the sections in the CXML and the sections in Typefi or the sections in the templates, and I started making all those connections and I was like, wait a minute, I can make my own templates. And so I started messing around with that, and I had zero knowledge of InDesign so that was a learning curve.

(13:02):

I would go to our production department and be like, how do I make a text box? And how do I name it? So it was a collaboration between myself and our production department, but just not knowing what I didn't know and that ability of youth to just move forward anyways, we were able to start generating CXML straight from the PUB app and sending it straight to Typefi without needing the NLM. Again, the NLM was great for our content, but this is stuff that's coming straight from our database. So the title page is just the title, the editors, all that stuff. So I write that out just as CXML and I submit it straight to Typefi and we get an InDesign document, and it was great. Same thing with the TOC. I'm able to write out the TOC straight from the spine file information and generate a TOC in seconds using Typefi.

(14:00):

So the flexibility of Typefi and how easy it was to just read documentation and then do it, I don't know that we really even had to consult with Typefi very much. There was a few hangups where we were just, send an email off and we're like, hey, we're stuck here, how do we do this? And they send us an email back, like, hey try this. We go and then it works. So I wasn't that surprised at the time because I hadn't

had any bad experiences yet, but in the years since then I've come to realise just how easy and smooth that was and how much that's not always the norm.

(14:41):

The problem was we still had a very print-centric solution. Again, Justin comes to me and he's like, hey, EPUBs are a thing. We need to start making EPUBs. And I'm like, okay, well how do we make EPUBs and what are EPUBs? This is probably 2011 or 2012. EPUBs are still a relatively new thing. And again, with the mindset of a youth of just moving forward and not necessarily caring about what you don't know and just fearlessly moving forward, I was like, okay, I can do that. So I started with, I took the XSLT transform from our Typefi and I was like, this is not that far from HTML. So I just went through the XSLT and taught myself as much about it as I could. Changed a lot of the tagging to be HTML, eliminated some of the extra tagging that we were using for InDesign, and I was able to generate basic HTML from the XSLT, so transforming NLM to HTML.

(15:55):

Then I made a Stack Overflow post, which anybody here programming is very familiar Stack Overflow. And I said, hey, I need to make EPUB. It needs this very specific zip format. I don't know how to do this and can someone please help me? I worded it a little better than that, but that's essentially what I did. And I got a response from the developer of one of the ZIP APIs and he said, hey, this sounds great. This is really new to me. I don't understand it either, but this code should meet your specs. And so I modified his code to fit into mine and lo and behold, we were just making EPUB. It was that easy. Put all the files in the right place, zip it up with some specific headers, and all of a sudden you have EPUB. So we did bookmark, sorry I wasn't moving the slides. We do website on our, we do HTML on our website.

(16:59):

We do HTML in proofing, and we do HTML for the EPUB. It gets to be a lot because they're all generated in slightly different ways. The EPUB is from the NLM using a transform. The website is, it just happens on the website dynamically through a bunch of find and replace and RegX, and in the proofing HTML happens through a C# transform. One of the big things with our EPUB was the math. Again, just trying to figure out the math. I found out, hey, I can automate InDesign. Why not try and automate Illustrator as well? So I found a way of combining some JavaScript that I had written with our graphic designer to convert our EPS files into SVG files using Illustrator. So that would run on, I'd run a script on people's computers to generate standard vector graphic. It's a way of generating graphics that will scale with size in the EPUBs because other than that, we're using PNGs, which, having math equations written out in PNG just doesn't work. It gets too grainy and you can't see it. I apologise for rambling. Like I said, it's been a long time. So I'm going to turn this back over to Justin to discuss the Typefi upgrade. Sorry, that is essentially where we are now. We have a lot of automation in place, but a lot of it's in-house and a lot of it's a bit hacky with Illustrator automation, InDesign automation, and the such.

Justin North (18:56):

Don't worry, he'll be back momentarily.

(18:59):

As Clay mentioned, throughout the years from 2009 onward, we've been doing a lot of gap filling, a lot of filling, like I think Chandi or Caleb mentioned earlier that workflow management wasn't something that Typefi had in 3 and 4, which currently we're still running 4 and we'll get into that in a minute. But a lot of the things that Typefi couldn't help with and our content management workflow system failed with, we had to pick it up. So that's what we did and that's where Clay learned so much about how to easily interact with Typefi, which is very easy to work with. So when looking at moving from our current environment to something new, obviously Typefi 8, we had to do a pre-implementation assessment. So we looked at

the pros. The pros, the stability. Right now we have a system that runs and we don't as an IT department need to manage it very much.

(19:51):

It just works every day inside and out and we can work on other items. Flexibility, once again, talking about proprietary systems and us filling the gap. We can write code so we can pretty much do whatever we want, which is, Clay wanted me to say it's a pro, but it's also a con. And then the cost. If everybody remembers when Adobe moved to a subscription model, everybody kind of said, oh, how are we going to move forward? It's going to be very expensive. Well, we never did, but we are going to now. So that was one of the reasons. Cons obviously had mentioned already maintenance. Every line of code you write, you are now on the hook to maintain and you can only accrue so much code debt before you have to pay that debt. And we're to the point where we're writing a lot of code, which is great, but at the same time now there is that maintenance side.

(20:45):

We didn't have a solution for math, obviously something we need to look into. And then multiple sources, which when I talk about multiple sources, we'll try to cover that in the next slide. And Clay will also discuss multiple sources moving forward. So once again, this is a pre-implementation assessment. So as I had mentioned before, we're moving and this is a huge jump, which we don't see as an upgrade as much as a completely new implementation moving from version 4.1.3 to version 8. Which, I don't think anyone here is using 4, is that correct? Thought so. Alright. So although we do have that stable environment, as I mentioned before, we've identified there are major areas in Typefi that have grown over the years, eXtyles, and most importantly the Adobe product line. Really starting to run into issues with speed and latency on the personal machines.

(21:40):

The proprietary code maintenance, just go into that a little bit more. The EPUB creation that Clay talked about earlier, although it works great, it's something that now Typefi can do for us. That was not the case previously, but now when we look to try to take each of these gap fills and try to push them off into Typefi using their professional services and the features in the new systems, that's something that we decided was a key area. Clay mentioned the table of contents, previously he was creating the table of contents. Everything that Clay mentioned, almost everything that he mentioned earlier that we have taken or he has come up with a solution for, Typefi is now going to be the solution for it. And one of those things is the table of contents, working with Jamie and the team on creating the table of contents, the detailed table of contents, all based off of other items, not something that we have to create individually.

(22:31):

The index, obviously I think a lot of people have been using Typefi for that, we haven't. But it's yet another thing that they'll be taking off of our plate. The EPUB creation, once again, something that we will now get out of the box and will be more in line with the style of our print product than what we have today. It's all customised and it goes into the PDF management. Clay mentioned the spine file that we are using, which is basically an InDesign book file, but something that actually works for books of our size. I know Clay had mentioned the size of some of our books, but we have 10 volume books and that's 10 volumes and each volume is 800, 900 pages. So we have some pretty hefty books. And then the differing HTML that he mentioned. We're creating HTML for five different reasons all in a different way. If we now using Typefi, we're going to get HTML from one source and we're going to be able to push it out in all directions. The math, which I already mentioned and multiple sources, and this once again is something that Clay is going to be able to describe a little bit better. Multiple sources is quite vague, but I think Clay will clear it up here in just a second.

Clayton George (23:53):



Alright, so the original Typefi implementation was great. Getting the InDesign in a almost perfect state was a huge time savings. Having it typeset all the figures in relatively the correct place was great. It was just, in CS4—which there's a reason no one's using it—it's just never, never going to be perfect. There's just not enough tools and not enough layout engine available to have it come out of Typefi perfect. And I don't know that it was Typefi, I would blame it on CS4 just because it was just so limiting it. Now, we in our latest upgrade, we're very close and I'm now to the point where I'm confident we're going to get there. We're going to have a hundred percent typeset PDFs coming out. We're going to have the InDesign document along with it. And in the very rare case where there is a small, hey, this table needs to be resized, hey, this figure needs to be bumped a little bit, whatever. But I think that will be far more rare than it has been in the past. So we're getting PDFs as our output instead of InDesign.

(25:14):

And this is a big thing. So I spoke about all my automation and it sounds great. I'm able to create these TOCs, able to create the spine file. I'm able to generate all of these different things, able to automate Illustrator. The thing I didn't mention about the Illustrator automation is that it basically takes control of your computer and makes it unusable while it's processing images. So that is an entire person's PC that is unusable, for some of our books, it was two to three hours where it's just sitting there churning through Illustrator and just every 15 to 20 seconds just taking control. I could not get Illustrator to run in the background. I don't know, maybe if I would've talked to Typefi or Adobe, but I just couldn't get it to run in the background. So a lot of the InDesign automation was the same way. It takes control of your computer and you can't really work. Now with the resource allocation, we're pushing almost all of that work off onto the Typefi Server. So the people will be allowed their own PCs and Typefi will do almost all of the work for us on their own Server out of everyone's hair, which is great.

(26:32):

We have a lot of electronic benefits. We're getting, again, typeset PDFs, which we'll be turning into eBooks. We get EPUB and HTML instead of XML as our output in our electronic, we get MathML. This is a big upgrade from eXtyles. We're upgrading to BITS and JATS, which I didn't mention here, but we're also adding MathML directly into our BITS and JATS instead of having NLM here and then exporting the EPS files separately, which was always a headache to keep everything in sync. And again here, resource allocation, like I said with the Illustrator, it's just so frustrating to not be able to figure that out. Every six months I would go and I'd spend four or five hours researching how do I make math better? How do I make it without clogging up people's computers? How do I do this? And I would just never come up with satisfactory answers until we got MathML and I even tried generating MathML from our EPSs.

(27:43):

I could never get a hundred percent. It would do 95% of the images and 5% would fail. And when you're doing things at the scale that we're doing, 5% failure is unacceptable. It's like we might as well just not even be doing it. So again, we're pushing all that work off onto eXtyles for the MathML. And then Typefi is able to use the MathML in our PDFs and in our EPUB, which the MathML in an EPUB is a world of difference when it comes to the large equations that we have and the ability that you can zoom in on the equations, it scales with the text. It's just a whole new world. So all of that stuff is great. I never really, I'm not a designer, I don't, I'm not a very artistic person. So all that stuff was done at the request of our production department, make things look nice, everything like that. Justin is better with design. He's like, hey, this isn't acceptable. You can't see this. I'm like, yeah, but you can see it here instead. And he's like, well, you should be able to see it here.

(28:57):

But this is the thing that always bothered me. We ended up having multiple sources. So this is back in the CS4 days. We would have the print workflow, which is on the left up here. We'd have the end user would generate our NLM XML by exporting it from Word. Then that would get converted into HTML in the

multiple places. EPUB being the main one I was concerned with. So I would generate that. Then we would have the NLM go off to Typefi. Typefi would generate the InDesign and we would get a PDF. And in the print workflow, the NLM is the true source of the workflow. Everything can trace itself back to the NLM. The problem is our production department is again doing small amounts of typesetting. They're not tracking what they're typesetting, but they're doing small amounts of typesetting in the InDesign phase.

(30:00):

So then we get to after print changes, which happens far more than any of us would like to admit. We get authors come back and say, hey, this paragraph was written wrong, we need you to fix this. It doesn't matter how much we think they should have done it earlier, we're still on the hook to do it. So we come in and the end user can't just modify the NLM and send it back off to Typefi because they've made changes to the InDesign in the meantime. And they would have to go back and make sure they make all the same changes again. And that's not a guarantee. We could end up chasing our own tail in that situation where we send it back and the author's like, hey, no, this thing changed. Why did this thing change? I didn't ask for it to change. So what we end up doing is the end user modifies the Word document, re-exports it to NLM, and that is the source for the HTML. And then the end user also goes into simultaneously the InDesign document and has to make the same changes there. Hopefully the same changes. Then that gets exported to PDF. So here is the problem, is that we have the NLM is now the source of the HTML and the InDesign is now the source of the PDF. So we now have two sources that we're now maintaining simultaneously. And the person that, the thing that is updating those sources is a person. And people are, from my experience, the most fallible part of a workflow.

(31:38):

So how do we fix this with the new Typefi upgrade? So we have the end user generates BITS and JATS now because we've come to the 21st century and they are sending that off to Typefi. Typefi then generates a typeset InDesign document, generally does not need to be modified. I'd say 99% or maybe more just does not need to be touched. It looks great. But if there is a change, they can modify that InDesign. Then it goes to Typefi, back to Typefi, we send the InDesign document back and we get all of our outputs from Typefi using the InDesign as the true source. So we have HTML coming from Typefi in the form of EPUB, and then also the EPUB being torn apart into the individual HTML, that HTML going to the website, going to our proofing system, going to whatever. We just use the EPUB that Typefi gives us, I no longer am on the hook to generate EPUB.

(32:47):

And then we also get our PDFs. So we generate our eBook from that. We generate our printable PDF from that. And you can see here, it's just a nice linear transition. I would imagine 99% of the time, even after print, the end user makes the change in Word, sends it through the BITS and JATS, and we redo this entire workflow so that the NLM is, or the BITS and JATS is the true source, single source of our content. In the event that the InDesign does need to be modified, we still only have one line that we're following and there's only one source. And that is, in the end, Typefi is the source for all of our outputs. And I'll transition back to Justin.

Justin North (33:38):

So one of the things I think that everybody wants to know is, what output changes happened? We already talked about the ways we've benefited from Typefi when it comes to streamlining a process or how we're going to benefit in the future, but everybody wants to know, how did they help your bottom line? So looking back, I know that Chandi mentioned it was 2007, but I don't know that the implementation actually finished until 2008. So I took my numbers from 2008. And in 2008 we published 145 books and 108 journal issues, which included 551 journal articles.

(34:15):

After the implementation, we ramped up to what came up to about an 80% increase in book output. And like I had mentioned before, our books are pretty big, so that's a good number. But the real increase was the journal articles, and I can kind of explain why that was the case. But that increase was 230% over the first year. And the reason that the journals jumped so high, the journal articles and not the books, is because you can acquire journals at a faster rate. Books, you need to really plan ahead. Books take much longer to produce. Some publishers take upwards to two years to produce a book from acquisition to actual publishing. But journals will show your true number of increase in output. And that right there at 280, 230% increase over such a short period of time is impressive. And 2017 is where we saw a substantial jump.

(35:11):

And the reason I picked 2017 was because right around the 2011, 2012 time period, like Clay had mentioned, we were really starting to start to fill these gaps because we had tried the content management system, we tried the workflow stuff, we tried to use what was invested in, but it just wasn't going to work. So that's when we started to manage things ourselves. And I don't think it was until around 2017 you'd see the benefits of all the investments. And that brought on an 180% increase in book output and a 325% increase in article output, which you see the increase for the articles was 2,348, somewhere around there. Which brings us to present day. So right now we've done a little over 660 books this year, 152 journals, and I will explain that in a minute, which is 1,150 articles. And you said, wait, why is that the case?

(36:02):

What happened? Did something slow down? Yeah. Well no, the industry changed, and I don't know if anyone else here is in the research publishing industry, but journal articles have changed drastically. There's no subscription based journals anymore. Everything's moving towards the open access model. The one thing I think makes IGI Global unique is we're one of a very few journal producers in our market that has 100% every one of our journals is gold open access. So we've completely switched everything, which that's why the numbers dropped because not every publisher is doing that. Some publishers, they only do journals. They can't afford to do that because that's their livelihood. It's not ours. So we've decided to go a different direction and try to push that open access model for the industry. So that's why you'll see a drop there. The next slide is what we predict or what we can envision based off of what we've seen with the new upgrade with all the publishing, I'm sorry, all the professional service things that we've seen thus far.

(37:08):

We project that we'll be up to 850 books per year. And if you think about a publisher that has 50 employees, maybe what, 8% of that is production staff, the rest is marketing and other things. Our production staff is very small. So these numbers, I think, truly do speak volumes not only to Typefi and the automation they've been able to provide us with, but also where we're heading in the future with our integration with Typefi. So once again, from the increase in our acquisition efforts and then also our current Typefi upgrade to 8, we think that our relationship with Typefi and the automation they're going to provide us will get us to, like I said, where we'd like to be. That pretty much sums up everything that we had. Thank you all for your time.