23th Symposium on Theory and Practice of Shipbuilding SORTA 2018
27 - 29 Sep 2018, Split, Croatia

Out-of-The-Box Dynamic, Distributed Information Sharing Technology in the Marine Industry: A Case Study

Nick Danese[a,*], Stéphane Dardel[b]

a NDAR, 150 Rue de Goa, 06600 Antibes, France
b NDAR, 150 Rue de Goa, 06600 Antibes, France
* ndar@ndar.com, +33 4 9291 1324

Abstract

Advanced information technology is nowadays commonly *available* and, more to the point, taken-for-granted and used all the time by virtually everyone in all aspects of our normal lives, transparently and mostly unknowingly. "Domotics" were implemented years before Big-Data and IoT became fashionable terms in the marine industry. Smart technologies and advanced sensorial robotics became common place in factories while in our world battalions of 3-axis cutting machines, flat panel lines and pipe fitting machines, unquestionably innovative when first deployed decades ago, still run alongside just a few modern, multi-axes profile shaper and cutter robots. Advances in hardware manufacturing, software and in information technology area are symbiotic and have blossomed into a self-sustaining evolutionary chain reaction. The marine industry uses only some of the vast wealth of very common and equally advanced technology that permeates our world. The very many factors that continue to sustain this anachronism include a widening generation gap, resistance to change at all levels, misunderstanding of the great opportunities offered by automation that serves people, an inexplicable refusal of simple, agile strategies and solutions, etc.. Making information *available* to serve all, people and processes, is the conjunctive tissue of the complex organism that is every industry and society. Focus of this study, solutions and answers are readily *available*, today, *off the shelf* and *out-of-the-box*. The technology and strategy adopted by SSI, creators ShipConstructor, is reviewed and case studies are presented.

Keywords: EnterprisePlatform, ERP, NDAR, PLM, ShipConstructor, SSI

1. Technology becomes evident when it is not there.

The electric kettle that turns itself off when the water that it is heating comes to a boil is a still common example of early automation at the service of humans. The consumer of the hot water is not concerned with the kettle and keeps otherwise busy waiting for the kettle's "mission accomplished". This is and evident example of useful information being made *available* to the consumer. A modern version of the above could be, for example, an energy pulse triggered by the water reaching exactly 100 °C, which turns off the gas flow in the cooker via an RFI device that actions a heat-charged, battery driven valve and sends a sms to a cell phone. The components of the process change, the process and the goal of making information *available* do not.

The medal has two sides, though. If the thermostat were to fail the kettle would perhaps not stop heating the water and possibly burn itself with nobody knowing until too late. Information about heating process information would be accessible to the consumer watching the kettle and, in this case, easy to find. Clearly, the information being made *available* is a preferable option.

The electric kettle burning itself out because the thermostat failed makes the kettle's technology evident *after* it went missing, thereby highlights the insufficiency of this information strategy. Back-ups and alternatives are easy to find in the case of the kettle, but damage limiting remedies are generally not so easy to come by when it comes to the much more complex ship design and ship building processes. Yet, a treasure chest full of effective tools, techniques and processes designed specifically to help prevent such situations is *available* via software running on standard hardware already in place in every office and most homes on the planet.

Much drama in ship design and ship building is caused by not having spotted the calculation error early enough, or by having serviced the welding apparatus in good time, or by not being able to react and quickly adjust the working schedule disrupted by the broken welding apparatus, or . . . Ironically, information technology is showcased here not by its disappearance due to failure, but by it not being used in the first place, intentionally.

2. Out-of-The-Box Dynamic Information Sharing.

All the technology, strategies, products, etc. reviewed here are *available off-the-shelf, out-of-the box*, today. With EnterprisePlatform, SSI is one of several companies having embraced the *Out-of-The-Box Dynamic, Distributed Information* philosophy, if perhaps the most advanced in this field in the marine sector. Designed to complement the definition of a fully developed PLM system that includes ERP, DMS, ECDM, HR, etc., EnterprisePlatform makes a huge wealth of information *available*, delivered to the desktop of each consumer (in the case of people), fully custom formatted and containing exactly the specific content requested and planned for that moment in time.

The paradigm implemented by the EnterprisePlatform is that of *available*, not just accessible". In today's terms, this is easily explained by comparing a few information "technology" heavy weights: Google, Facebook and Amazon.

- Google and other search engines only make information *accessible*, but then only potentially so: when looking for something one tries to formulate the best possible search request and even then it is not certain that the sought result will be found despite it definitely being there. Try launching the same search a few minutes apart, or searching for a "man with mammoth" image . . .
- Facebook and other data subscription services, deliver what you requested to your desktop, thereby making the information *available*. You specify what you want (as allowed by Facebook) and, when *available* it is delivered to you directly without asking or searching.
- Amazon, and other custom advertising services make unsolicited information *available*, based on their own interpretation and assumptions of what the consumer *might* want, generally generating useless noise – although AI seems to improve on this situation.

Back to the *available* paradigm, the ability to bespoke, useful data sets and deliver them to numerous consumers with different requirements, at the required times, in different formats, etc. from a multi-source and multi-authoring environment is paramount to distributed, relational decision making in a most efficient fashion.

3. The Distributed Data & Information Relational World.

Data & Information have always been generated by multiple sources, connected and disconnected and have forever been related across the spectrum of "everything". Sources can be people, processes, events, etc. Regardless, the adagio "a butterfly flapping its wings will create a storm thousands of kilometres away" tells us that pretty much every component of a process will influence all other components, and so will every process, in a *many-to-many* fashion. To concentrate on only one or a few components might improve this or that components, but that does not necessarily follow when it comes to the overall process.

Over the years, many attempts to make sense of such a non-sequential, many-to-many environment have failed. The reason is often that rational strategies were applied in a rigid framework based on a cause-&-effect lay line or, at the antipodes, in generating useless data & information floods which succeeded only in creating dysfunctional noise.

The marine industry pays little attention is paid to what happens in other industries and the many life-size experiments such as "just-in-time", "LEAN", "Agile", etc., all based in sharing information dynamically and focused largely on opportunistically making the most of what exists and preventing issues. In retrospective, no one discovered the *one way to do it right*, perhaps very simply because *there is not* one way to do it right. In fact, it may very well be that a collection of connected, related, associated, “local” solutions to “local problems” underpins and composes a dynamic ensemble of intertwined solutions capable of tackling the non-sequential, many-to-many scenario effectively and, one step further, allow some level of rational management to be applied to an irrational collection of mostly irrational processes. The immediate analogy is the human brain which, unlike software, also has the all-indispensable intuition.

Several strategies aimed at tackling such problems have been proposed over time, more recently various flavours of swarm theory (and practice) [1]. One very early and eye-opening vulgarisation thereof is found in Michael Crichton’s PREY (2002). This cautionary tale very clearly resumes and connects in many-to-many fashion artificial life, emergence (and by extension, complexity), genetic algorithms, agent-based computing, population dynamics and host-parasite coevolution, evolutionary technology, nanotechnology, genetic engineering, etc. under a somewhat disturbing (in the novel) distributed artificial intelligence umbrella.

4. Going with the flow.

Although swarm theory in not quite yet commonly applied in its most complete and advanced forms, the concept is put into practice every day by people, from individuals to teams to entire societies. But, if on one hand the human mind is perfectly capable of analysing, resuming and plotting a forward course, on the other *input* is required to support even those endowed with the most powerful intuition.

Input, direct, fractal or analogical, is in its most general definition, the DNA of problem solution and can be thought of as a combination of interconnected data and information. Ironically, despite being overly abundant, input is not exploited anywhere near as much as it is easy to find. One main culprit is human nature, rarely rooted in technology or even just technical grounds.

Going with flow carries at least two, fully contradicting meanings, both found in every facet of everyday life, personal and professional. Does it mean keep on doing what everyone has been doing forever or does it mean to grasp a fleeting opportunity and break out of convention ? A nice example is that of Bilbo Baggins, in JR Tolkien’s book The Hobbit (1934):

- the night before Bilbo Baggins was to set out on an “adventure” he categorically dismissed the idea: the convention flow
- the next morning, Bilbo Baggins chased Gandalf and the Dwarfs to go on an “adventure” and even rode a horse (backwards !): the dynamic, creative, opportunity flow

To dynamically adapt processes in a successful way does not require starting from rational, scientifically proven and absolute positions. In fact, the *dynamic and adaptation* components somewhat contradict the very nature of “certainty” and “known”. Corollary and engendering at once, innovation and creativity fuel the intuitive essence of dynamic adaptation. The transformation of intention into action and achievement of success are made possible by an equally dynamically adaptive application of many different techniques and strategies (*just-in-time*, Agile, LEAN, etc.) brushed on a weave of swarm-type threads. The flow of data and information acts as both paint thinner and fixer at once.

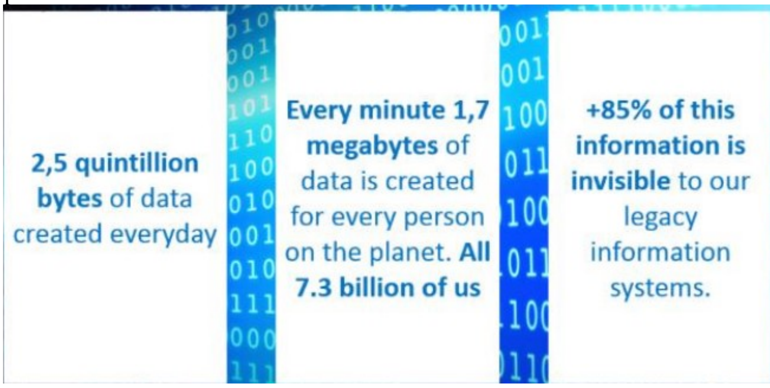


Fig. 1 85% of currently created information in invisible to our legacy information systems [2]

5. Data & Information: useful *vs* useless.

Data and information are of value only if they are understandable, applicable and *available* at the right time (which can be at various times, not just one specific instant), to the requiring consumer.

Even in digital form "understandable" encompasses a vast array of meanings, going from spoken language to encryption, all born of authoring and formatting. Authoring and formatting carry a legacy of causing data & information to be, at best, only partially useful, due to the traditional insistence of converting one unique format into another unique format, generally very hard if not impossible due to fundamental differences in the definition of the data being converted and/or to the insufficiency of a given data set.

5.1. Understandable.

Understandable is not easy to explain, let alone understand, because it means something different to all involved in the overall process: people, machines, processes themselves, etc.

If insisting on standards; failure to convert between them can sometimes be mitigated by combining more than one data set and/or adding meta-data, thus provide a data set exploitable by "another" standard. In this respect, the analogy with "swarm" thinking is fascinating, but still considered a mermaid by the marine industry.

Perhaps, much trouble could be avoided by accepting that "standards" are like Don Quixote's windmills and by adopting an *Esperanto*-style, multi-authored, rich in redundant variables yet congruent and valid-compliant data sets.

Along these lines, XML (eXtensible Markup Language) is one of various options *available* to us to collect heterogeneous, possibly redundant and yet complementary data and information, thanks to its many virtues:

- it is human readable: no need for interpreting, decoding or decryption
- its contents are self-explanatory: no need for hard to maintain documentation
- it is infinitely extensible and supports any sort of data and information
- it is writable by any authoring tool, in whichever "format" is deemed appropriate
- it complements other types of data set, such as CAD, "neutral" file formats, etc.
- it intrinsically supports evolutionary data ("versions")
- etc.

In other words, XML contributes to evolve and distribute the standard-agnostic Definition Model [3] .

5.2. Applicable.

Even easier, *applicable* simply means that the data & information set is *what* the consumer requires, however *formatted* and *up-to-date whenever* required, perhaps composed of several data sets that come from several authoring sources and stored in several locations and formats. By extension, *applicable* also means that the data set is of the "right" version, according to the requirements.

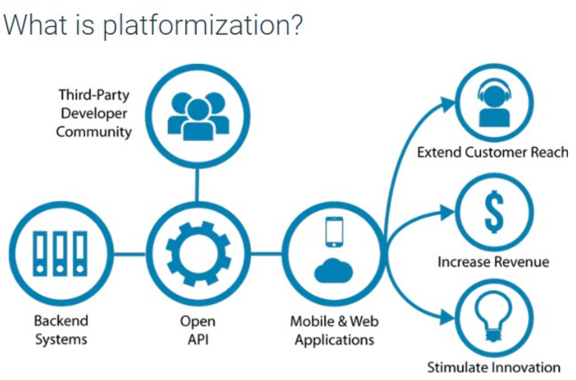
It would be easy and tempting to discount the above by chanting the glories of relational databases such as SQL and Oracle but, powerful tools as they are, these require proprietary interpretation schemas and tools before their content becomes just *accessible*, let alone *available*. On the other hand, the high level of complementarity between relational databases and XML is instrumental in forging a very effective strategy, notwithstanding the great data sharing tools to come.

5.3. Available.

Enter the troublemaker: *available* neither common nor a standard strategy, in too many cases not even possible. *Available* simply means that consumers' data & information requirements are satisfied. PLM processes are meant to make data & information *available*, but remain totally dependent on authoring tools to make *their own* initial data & information *available* and thereby are more a sort of post-processor. Authoring tools are the primordial source that will feed other authoring tools, data, information and processes, much like the many rivulets that become streams and that eventually make the river where that wears obstacles down to achieve the least-energy course to the sea *at that time*.

6. The Many Sources of Many Truths.

At a time, the *single source of truth* was to best support holistic strategies [4], and the idea of a distributed, potentially redundant collection of truths generated by several authoring tools fuelling the "Synchronized Shipyard" engine was still embryonic. Platformization, big data, Internet of Things, data driven models, digital twins first and then functional digital copies, etc. have changed the theatre and foster the co-existence of many truths, ranging from complementary to contradictory, yet all equally valid for some purpose x at some time t. In fact, all truths contribute to one, dynamically adapting "super-truth", never to be taken as a whole at any one time but always containing all the components required to create the one truth valid at that one given time.



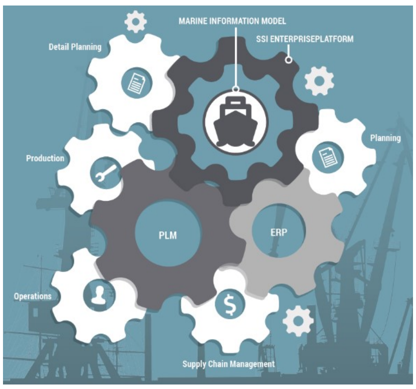


Fig. 2 Platformization: to create a marketplace/environment at your service
Fig. 3 The Synchronized Shipyard

Which truth is required at which time is defined by the consumer, dynamically. In many ways this is what a PLM system is tasked to do but, again, it works downstream of the authoring tools. The authoring tool then becomes responsible for the production and delivery of useful source data and information. If the *deliverable* is contributed to by more than one authoring tool, all must contribute useful source data to the next consumer.

The contrast with how little most authoring tools allow today is evident, as is the difficulty in setting up a PLM system to orchestrate such data & information flows. One answer is to go back to the basic paradigm and work with discrete data & information sets which are managed and validated by their respective authoring tools when generated. The XML data structure lends itself well to this approach, as it can easily include recognisable "validity" parameters exploited very effectively by PLM.

7. The Consumer.

The consumer is king In this scenario, be it people, machines, processes, etc. individually or in combination. More deeply, the first consumer is possibly the mechanism that delivers the source data to, for example, the XML file or PLM environment. Several aspects of this first path are very relevant and deal directly with the authoring tool itself:

- is the data *available* from storage ?
- is the data *available* only when the authoring tool is active ?
- how to tell if the data in question composes a valid, useful set at the time of issue ?
- at the time of consumption ?
- etc.

Data mining is then the first, crucial waypoint in the process of making data *available*.

8. SSI's solution: the EnterprisePlatform.

SSI, Canada, developed the EnterprisePlatform, a Representational State Transfer API (commonly known as REST) based environment that "gathers, formats and provides meaningful, actionable information for distribution and consumption across the entire organization". The mechanism is complex behind the scenes, but disarmingly simple to use.

8.1. The goal.

The goal is to make a useful, customized data set is composed from the extended ShipConstructor + Autodesk environment (but not only) made *available* to the consumers. Some of the constraints are:

- each consumer may require several data sets to be made *available* at the same of various times
- each data set may be required by several consumers at different times, validated by different criteria
- the same data set, or just portions thereof, will be required by different consumers in different formats and/or representations (tables, drawings, graphs, etc.)

Let us keep in mind that the consumer need not be a person, it can a database (ERP, PLM, DMS, EDM, etc.), a machine (robots, etc.), a process, an action, etc.

8.2. The scope.

EnterprisePlatform operates over an extended environment, that includes but is not limited to:

- the MIM (Marine Information Model): the ShipConstructor SQL project database
- Autodesk products and their files, other databases
- any executable software, its scriptable commands, exposed variables, run-time data and files
- any *available* data and information stored anywhere within reach, directly and indirectly
- other REST-based systems
- etc.

Ambitious as it might sound, it is an achieved scope of operation that complements existing scenarios and directly supports other systems, thereby contributing to an evolutionary enterprise process.

8.3. System Architecture.

In the case of the MIM, the EnterprisePlatform (EP) uses REST to manage a bi-directional data and information flow. EP is composed of two main building blocks, discussed below:

- Server: fully autonomous and automated, made of the Information Hub, Connectors, Operations and Scheduler, based on one or more "servers" and one or more "nodes"
- Client, effectively a self-sufficient server and node, runs Operations, locally

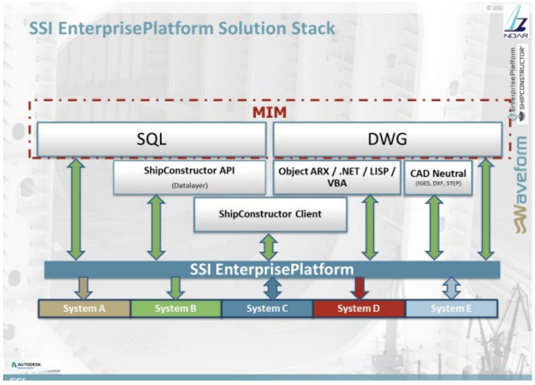


Fig. 4 EnteprisePlatform communication schema

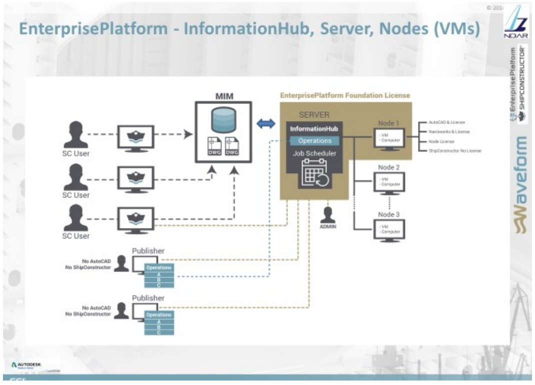


Fig. 5 EnterprisePlatform functional layout

8.3.1. Operations.

Operations are clear, text format XML files that can contain anything, from data to connections to other systems, to instructions to interact with other applications, including launching and executing, etc. The following example from the Operations.xml file is *available* to all EP 2019 users *out-of-the-box*:

```
<!--@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-->
  <!--Profile Part DXF for a Profile Cutting Machine with Bend Markings-->
  <!--@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-->
  <operation id="Profile Cut Dxf With Bend Info" appliesto="parts" maxgrouping="50">
    <call procedure="LogLauncher">
      <string>Profile Cut Dxf With Bend Info</string>
      <string>OperationStart</string>
      <string>Operation started</string>
    </call>
    <define name="operation.timeout">
      <invoke function="GetOperationTimeout">
        <property name="list.ids" />
        <int>10</int>
        <int>800</int>
      </invoke>
    </define>
    <define name="alignMarkStyles">{CASSETTE}</define>
    <acad timeoutinsec="${operation.timeout}">SCEPGENERATEPROFILECUTDXFWITHBENDINFO
${list.ids} "${alignMarkStyles}" "10" "1" "1" "50" "AC1015" "" "${rootdir}\Parts\ProfileCNC\" "NO"
"[PART_ID]"</acad>
    <call procedure="LogLauncher">
      <string>Profile Cut Dxf With Bend Info</string>
      <string>OperationEnd</string>
      <string>Operation ended</string>
```

```
    </call>
  </operation>
```

The Operations environment is open, documented and totally user configurable, *available* out-of-the-box to ShipConstructor and shared publicly via the SSI Nexus Forum. It is customary for most to quickly start developing their own Operations.

8.3.2. Connectors.

Connectors use Operations to custom manage a one- or two-way data & information flows with systems (ERP, PLM, etc.), machines (robots, etc.) and data repositories (SQL, Excel, etc.).

8.3.3. Server and Client components.

In the SSI realm all tools are *available* to all, tailored to serve the scope of implementation. In the case of EP "EP server" and "EP client" set-ups are distinguished:

- Server offers total versatility and autonomous, automated execution of Operations within or without Connectors, controlled by the Scheduler in a totally distributed architecture comprising one or more servers and nodes, running on physical and/or virtual machines, across LAN / WAN / Cloud. Node will host and run local sets of software are required by the assigned Operations (ShipConstructor, AutoCAD, Navisworks, MS-Office, etc.).
- EP Client runs locally, using local resources installed on that PC, physical or virtual, outside office hours, on-demand, etc.

9. Case Studies.

The following case studies provide a concise yet wide-spectrum cross-section of EP implementation, attention is called to the varied field of application and goals in the use of EP. While not mentioned explicitly, other marine software such as ShipWeight by BAS Engineering, etc. are part of the EP environment, as well as many other software systems.

9.1. Huntington Ingalls Industries, USA

HII design and build Navy ships, from small Corvettes to Aircraft Carriers. Following an initial business process assessment (BPA) and compelled by a fixed-price policy enforced by the main customer, the US Navy, an in-house "PLM Research & Development" team tasked with the identification of achievable, short-term milestones was appointed, integrated by specialists from SSI, ARCOS (an ERP/PLM implementation specialist consultant) and ARAS. A first 8-month development and implementation exercise brought on-line across the company in August 2016 several enterprise processes supported primarily by SSI's EnterprisePlatform & ShipConstructor, ARAS's ERP/PLM, Autodesk products (AutoCAD 3D, Vault, etc.), MS-Excel, etc. with a minor role retained by MARS. To better serve the ever-changing scenario, the on-going milestone cycle was shortened to three months and "foundation" subjects privileged. Very importantly, short cycles also allow to re-visit the architecture of the evolving process and adopt technological innovations.

9.2. Royal Huisman, The Netherlands

Royal Huisman design and build some of the most technologically complex and largest sailing yachts within the scope of a very short *contract-to-delivery* time, which requires significant support of the *many-to-many* process discussed in this study. EnterprisePlatform is instrumental in automatically reviewing data integrity and quality, reporting on project progress, continually updating the deliverable's archive and making useful, custom formatted data and information *available* virtually in real time to certain non-ShipConstructor consumers, in some case as often as every 15 minutes, such as the interior furniture and production teams. Implementation of EP was initiated by the CAD Manager, and subsequently taken over by a dedicated team member for a more systematic analysis of requirements and EP-based tactics and strategies.

9.3. Drydocks World Dubai

In recent years DWD have been particularly active in the production engineering and building of offshore units (platforms, jackets, etc.). Characteristically, welds are a highly regulated component of the build and a fully detailed weld model is built automatically with ShipConstructor for use by Production, Quality Control, Inspection and Quality Certification by Class. Weld data and weld information are shared bi-directionally via work-packages published from the model environment and a MS-Excel interactive interface on mobile devices (ex. iPads) by which field data about welds is input back into the Marine Information Model database to augment and annotate weld documents (drawings, tables, 3D models, status reports, etc.). EnterprisePlatform manages this process, as well as the publication of a variety of custom-defined 3D models for collaborative engineering in Autodesk Navisworks format, the supply of production data in custom format to DWD's ERP environment centred on their Oracle-based Mariner system, and re-input of weld status documentation. EP is managed jointly by the Engineering and IT Departments.

9.4. BMT Nigel Gee

A very early ShipConstructor user, BMT Nigel Gee design a very wide spectrum of vessel types: fast ferries, service craft, large yachts, etc., and generate production engineering documentation for some. EnterprisePlatform is used to manage deliverables of the same type but destined to different shipyards with different documental requirements, in particular contents, labelling and annotation. The Operations used for the purpose are effectively variants of a template, exploiting the unique ability of EP to support extensive, different, custom sets of deliverable documents in parallel. The effort is currently lead by a project manager.

10. Big data, IoT, Digital Twins and Functional Copies.

IoT (Internet of Things) is the common denominator of many developing technologies, and the overabundance of (big-) field-data is the Big-Bang at the start of the new universe of empirical-based prediction strategies. IIoT (Intelligent Internet of Things) makes two-way data and information streams even more ubiquitous. Exploitation and integration of IoT data is already totally supported by many, out-of-the-box, SSI's EP being one.

Consider the common example of an old, on/off type welding machine plugged into a current flow sensor connected it via LAN, serial, WiFi, gsm, etc. That day (planning) someone (HR) will use the machine on a known job as described by a drawing (CAD) using a given anode or MIG wire (consumable resources), etc.: via an *off-the-shelf* IoT connection current consumption, welding time, etc., become immediately *available* to the many consumers (people, dashboards, ERP, maintenance crews, planning, etc.) across the enterprise.

Universal, fully plug & play data flow management systems like the IIoT ISS by Alleantia, Italy, are *available off-the-shelf*, designed to be implemented in minutes and deliver very high Return on Investment by supporting progress monitoring, preventive maintenance, resource allocation and consumption, planning, etc.

Fig. 6 The Alleantia plug&play zero-coding industrial equipment integration technology.

Current machines provide full data streams and, in the case of portable units, store data on board for off-time download and analysis, for example the portable pipe clamp by FIPE, Switzerland.

Fig. 7 The FIPE portable pipe clamp stores operational and environment data made *available* via a USB port.

11. Conclusion.

An easy, overall analogy to make is that of a.

Dynamic Information Sharing Technology directly at the Service of Current Day Ship Design and Ship Building can be thought of as a universal, omni-present decision-making support environment that is *available* today, all the way to IIoT and providing extremely high and fast ROI. Deploying such technologies via a focused strategy in an Agile way is uncharacteristically easy, quick, and requires no specific software or hardware expertise. A simple review of macro- and microscopic enterprise processes is sufficient to start the virtuous cycle towards reduction, perhaps even elimination, of waste, errors discovered in production and the crippling costs and losses they generate.

A "swarm" strategy supported by XML files offers untold advantages in its flexibility, adaptability and open-by-design availability to integrate new technologies and philosophies and existing processes and databases, all the while directly and actively supporting the very complex *many-to-many* decision-making environment of ship design and ship building.

A few COTS (commercial off the shelf) software and hardware products have been mentioned with the express intention of prompting the reader to look and find the many more which already contribute to success in non-marine industries, and to suggest the ease of implementation and extraordinary ROI when bringing them back to our industry to improve ship design and ship building.

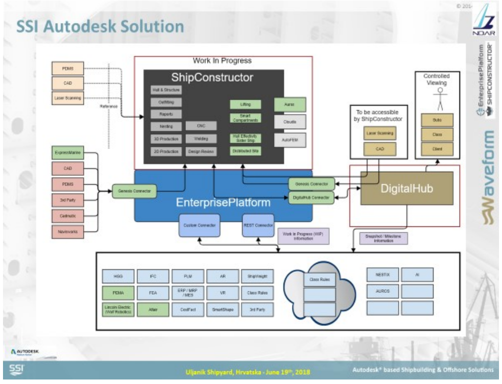


Fig. 8 The extended SSI & Autodesk solution built on the EnterprisePlatform

Acknowledgements

The authors wish to thank colleagues and friends in the marine industry for their kind reviews, corrections and suggestion : Volker Bertram, Lambertus Oosterveen, Siniša Relijc, Marko Tomic, Marcel Veldhuizen, and everyone else.

A special thanks to Denis Morais, for many, many years of inspiration, brain storming and sparring.

References

[1]COMPIT, various authors, 2012-2018

[2]COMPIT 2018, various authors

[3]DANESE, N., MORAIS, D.: “The Synchronised Shipyard”, Proceedings, ICCAS 2016

[4]MORAIS, D.: Waveform Blog, 2013-2018, several postings

[2]SENJANOVIĆ, I., TOMAŠEVIĆ, S., PARUNOV, J.: "Numerical Procedure for Ship Whipping Analysis", Brodogradnja 49(2001)2, p.171-180.

11.1. General

Authors should format whole document using the provided styles. The document is written in 12 pt regular Times New Roman, justified. Chapter titles are in 12 pt bold, numbered (*Heading 1* style). Sub-titles are in 12 pt italic, outline numbered (*Heading 2* style). Figure captions should be placed below, and table captions should be placed above. Pictures should be inserted “In line with text” (*Fig_body* style).

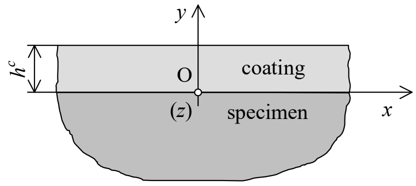


Fig. 1 Single picture example
Slika 1. Primjer smještaja jedne slike

Please ensure that every reference cited in the text by its sequence number in parentheses [1] is also present in the reference list (and vice versa). References should be listed at the end of paper in the order cited in the text. Authors are kindly asked to follow the style of given examples.

Equations should preferably be written in Microsoft Equation editor:

$$a = b + c \quad (1)$$

Table 1 Explanation of table content
Tablica 1. Naslov tablice

Column heading	Column *t*, m	Column *B*, %
An entry	1	5
Another entry	4	7

Acknowledgements

Collate acknowledgements in a separate section at the end of the article before the references. List those individuals who provided help during the research.

References

[1]HUGHES, O. F.: "Ship Structural Design", John Wiley & Sons, Inc., and New York, 1983.

[2]SENJANOVIĆ, I., TOMAŠEVIĆ, S., PARUNOV, J.: "Numerical Procedure for Ship Whipping Analysis", Brodogradnja 49(2001)2, p.171-180.

[3]DAWSON, C.W.: "A Practical Computer Method for Solving Ship-Wave Problems", Proceedings of 2nd International Conference on Numerical Ship Hydrodynamics, University of California, Berkley, 1977, p.30-38.